DM 6,50

c't magazin für computer technik

4

c't April 1986

Programmiersystem UCSD-p

Mit dem Segen der Post:
Modem-Box

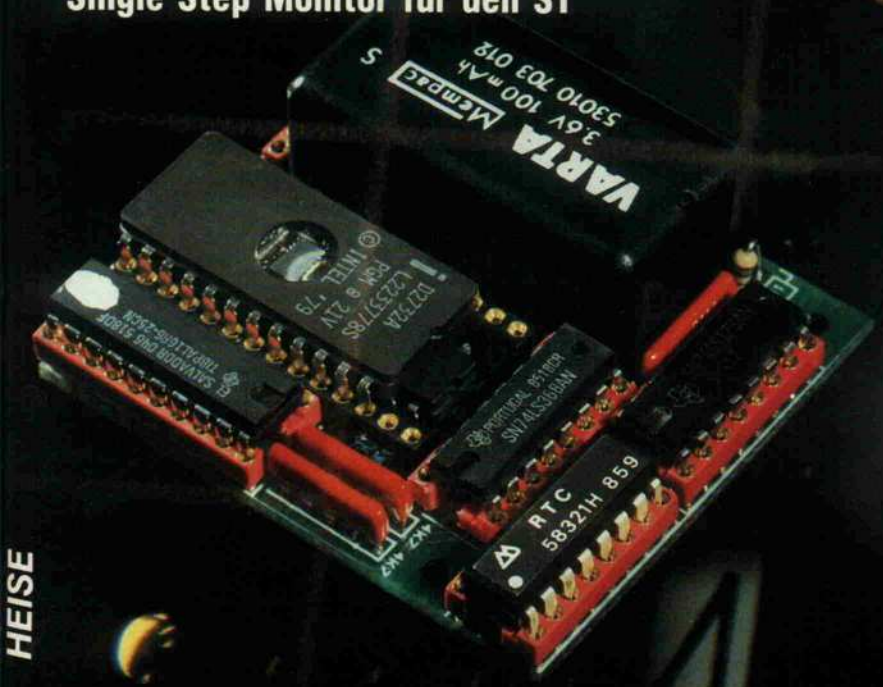
CP/M im Koffer:
Bondwell Modell 2

Das andere Zeichengerät:
Robotplotter Penman

Neu von Borland:
Reflex

ECB-RAM-Disk mit 1 MByte
WordStar-Patches
Numerische Integration in BASIC
1000 Programme pro Diskette
CPC unter CP/M mit äöüß§
Single Step Monitor für den ST

ö$ 57,– · sfr 6,50 · hfl 7,40



Nützliches Accessoire
für jeden Computer:

c't - Uhr

HEISE

PLANTRON

DIE KOMPATIBLEN QUALITÄTSCOMPUTER!

PT-16 AT
- CPU 80286, 6 MHz (80287 Option)
- 640 KB RAM (max. 3.2 MB)
- 8 Slots (2 PC/XT kompatibel)
- Color- oder Monochrom-Grafikkarte
- Parallele Schnittstelle
- Serielle Schnittstelle
- Multi-Controller Karte
- 1.2 MB Diskettenlaufwerk
- 25.6 MB Festplattenlaufwerk (bei PT-16 AT/20)
- Echtzeituhr
- AT-Tastatur
- Betriebssystem Ms-Das

PT-16 AT ..... DM 6998,–
PT-16 AT/20 ..... DM 8998,–

PT-16 XT
- CPU 8088, 4.77 MHz (8087 Option)
- 256 KB RAM (max. 640 KB)
- 8 Slots
- Color- oder Monochrom-Grafikkarte
- Parallele Schnittstelle
- Serielle Schnittstelle
- Game Port
- 2 x 360 KB Diskettenlaufwerke
- 12.8 MB Festplattenlaufwerk (bei PT-16 XT/10)
- Echtzeituhr, RAM-Disk, Druckerspooler
- Tastatur mit separatem Cursorblock
- Betriebssystem Ms-Das

PT-16 XT ..... DM 2998,–
PT-16 XT/10 ..... DM 4998,–

DATENMONITORE:

MBM 2233
- 12 Zoll Bildschirm (P 31), entspiegelt
- 18 MHz Videobandbreite
- BAS-Eingang (anschließbar an Color-Grafikkarte)

Grün ..... DM 298,–

MBH 2003
- 12 Zoll Bildschirm (P 31), entspiegelt
- 25 MHz Videobandbreite
- BAS-Eingang (anschließbar an Color-Grafikkarte)

Grün ..... DM 348,–
Bernstein ..... DM 358,–

SGM 1251 T
- 12 Zoll Bildschirm (P 39), entspiegelt
- 20 MHz Videobandbreite
- TTL-Eingang (anschließbar an Monochrom- und Color-Grafikkarte)
- Mit Schwenkfuß

Grün ..... DM 498,–

SHCM 1403 RGB
- 14 Zoll In-Line Bildschirm (0.31 mm Pitch-Abstand), hochauflösend, dunkel
- 18 MHz Videobandbreite
- RGB-Eingang (anschließbar an Color-Grafikkarte)
- Umschaltbar auf grün/bernstein Darstellung

Color ..... DM 1398,–

Weitere Informationen erhalten Sie bei Ihrem Fachhändler oder mit einem Händlerverzeichnis direkt von PLANTRON. Fordern Sie die neuesten 86er Prospekte an. Für Fachhändler sind die entsprechenden Unterlagen abrufbereit.
Auf alle Produkte 6 Monate Vollgarantie. Die angegebenen Preise sind unverbindliche Preisempfehlungen inkl. Mehrwertsteuer.

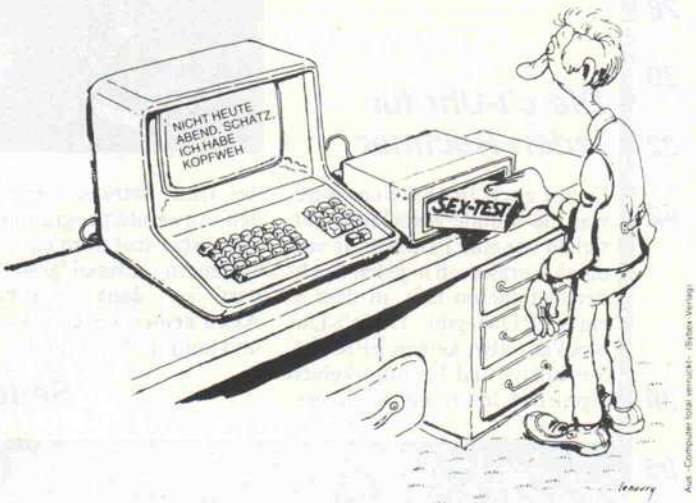
PLANTRON
Computer Vertriebs-GmbH
Höhestraße 28
6380 Bad Homburg v.d.H.
Telefon 0 61 72 / 2 51 88*
Telex 4 17 410 placo d

Liebe Leserin, lieber Leser,

ist diese Anrede nicht ungewöhnlich? Läßt sie einen nicht stutzig werden? Ein Magazin für Computertechnik, das sich mit nichts anderem als EPROM-Simulatoren, RSX-Modulen und ECB-Prozessorkarten befaßt, bildet sich doch tatsächlich ein, es gäbe auch nur ein weibliches Wesen in der Leserschaft!

Ist die 'Liebe Leserin' nicht eher der Versuch, das zu kaschieren, was schon längst offenkundig ist? Nämlich daß es keine so unangefochtene Männerdomäne wie die Computerei gibt. Ist die höfliche Erwähnung schon Ausdruck eines schlechten Gewissens?

Natürlich dürfen Frauen helfen, Computer zu verkaufen. Auf der CeBIT werden uns freundlich lächelnde Damen Prospekte zureichen und Computer-Modelle vorführen, die sich Lisa, Joyce und Vicky nennen. Im Amiga gibt es sogar zwei Chips, die Frauennamen tragen: Paula und Denise. Offenbar nutzen feinfühlige Werbepsychologen, die männliche Kundschaft im Visier, die Neigung mancher Computer-Hobbyisten, ein 'erotisches' Verhältnis zu ihrem Heimcomputer aufzubauen.



Aus „Computer total verrückt", Bertz Verlag

Die Datentypistinnen und Sekretärinnen in den mit Textverarbeitungssystemen ausgerüsteten Büros dagegen sind nur die Leidtragenden einer Technik, an deren Entwicklung und Einsatz sie kaum aktiv oder innovativ beteiligt sind.

Was hält Frauen davon ab, Nutzung und Anwendung von Computern mitzubestimmen? Das Klischee von der technischen Unbegabtheit und der mangelnden Fähigkeit zum logischen Denken ist doch längst als solches entlarvt. In allen Disziplinen gibt es hervorragende Wissenschaftlerinnen, aber wo sind die Informatikerinnen?

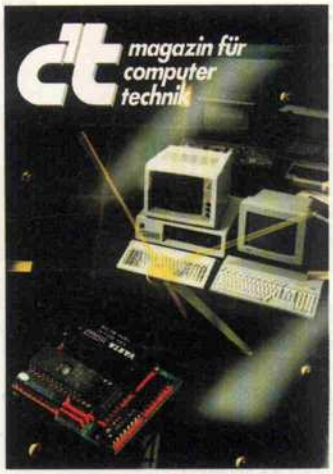
Vielleicht ist es ganz einfach wider die frauliche Natur, sich tagelang vor Tastatur und Monitor in Detailprobleme zu verbeißen und den Umgang mit Kollegen, Bekannten und Freunden zu vernachlässigen. Oder ist es das Primat von Zeitgewinn und Leistungssteigerung, unter dem die von Männern geschaffenen Programme stehen, das dazu führt, daß nicht nur Frauen Computertechnik als inhuman, unsozial und lebensfeindlich ablehnen?

Auch auf die Gefahr hin, mißverstanden zu werden, sei hier eine Computeranwendung erwähnt, die aus dem angedeuteten Rahmen fällt und die von einer Frau, einer amerikanischen Psychologin, maßgeblich mitentwickelt wurde. Es handelt sich um ein Programm, das, basierend auf einem Fragenkatalog, das Verhalten im Bereich der zwischenmenschlichen Beziehungen analysiert und angeblich in der Lage ist, ein sexuelles Persönlichkeitsprofil zu erstellen. Was immer das auch sein mag. Ganz egal, ob es sich hier um wissenschaftlich ernstzunehmende Hilfen oder um einen besseren Party-Gag handelt, zeigt dieses Beispiel zumindest, daß es auch lebensfreundlichere Anwendungen gibt als Tabellenkalkulationsprogramme ... bei aller Vorsicht.

In vielen vermeintlichen Männerberufen, in Politik und Wissenschaft machen selbstbewußte Frauen nicht nur ihre Rechte geltend, sondern sorgen auch für mehr Menschlichkeit. Warum wollen sie sich nicht auch an der Gestaltung von Computer-Software beteiligen? Computer werden in noch viel stärkerem Maße unseren beruflichen und privaten Alltag bestimmen. Liebe Leserinnen, sorgt dafür, daß es nicht nur männliche Computer sind.

Manfred Bertuch

(Manfred Bertuch)



4 86

Prüfstand

Schlüssel zur Welt 28
Mit dem Segen der Post: Modem-Box

CP/M im Koffer 30
Portable-Computer Bondwell Model 2

Der Mausplotter 32
Penman - ein neuartiges Zeichengerät

Lange Arme für Commodore 42
Computer-Interface von Fischertechnik

Praxistips

Patching WordStar - und kein Ende 39
Patch-Adressen aus vier WordStar-Versionen

Bitmanipulator 96
Bits setzen, abfragen, löschen beim Spectrum

Erste Schritte zur analogen Eingabe 111
Einfacher AD-Wandler am c't-ST-Userport

Software-Know-how

CPU-Gleichrichtung 52
Das Programmiersystem UCSD-p

Des Schneiders Kern 70
Interrupts. . .ticker. . .frame flyback. . .events

Veränderlich 90
Systemerweiterungen unter CP/M-Plus

Compiler selbstgeschneidert 98
Teil 4: Erzeugung von Stack-Maschinencode

Integrieren lassen 107
Numerische Integration in BASIC

Programme

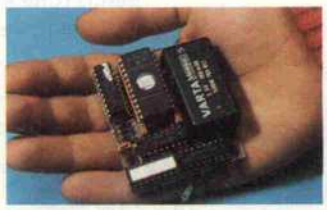
Deutsche Typen 60
CPC 464 mit deutschen Zeichen unter CP/M

Deutsche Steuern 64
Lohn- und Einkommenssteuerberechnung für 1985

Schritt für Schritt 82
Atari ST im Single-Step-Modus

Tausend Programme pro Diskette 84
Modulbibliotheken unter CP/M und MS-DOS

Titel



Die c't-Uhr für jeden Rechner

Damit auch Ihr Rechner weiß, was die Stunde geschlagen hat, stellen wir eine Echtzeituhr vor, die sich praktisch in jeden Rechner installieren läßt, in dem es ein EPROM gibt. Die c't-Uhr blockiert aber keinen EPROM-Steckplatz und benötigt keinen eigenen Adreßraum. Sie ist weder dem Betriebssystem noch den Anwenderprogrammen im Weg, aber trotzdem einfach anzusteuern. Einmal gestellt, bedarf sie dank eingebautem Akku keiner weiteren Wartung und läuft und läuft und läuft . . .

Seite 44

Schnelle Scheibe

Greifen Programme häufig zur Floppy oder muß man selbst des öfteren Tools von der Diskette nachladen, beschleunigt unsere RAM-Floppy ihr System. Unser Projekt für ECB-Bussysteme zeichnet sich durch stolze 1 MByte Speicherkapazität und einfache Einbindung ins Betriebssystem aus.

Seite 76

Pascal-Veteran

UCSD-Pascal gibt es auch noch! Wir wollen dafür sorgen, daß diese betagte Pascal-Implementation neben den Turbos nicht in Vergessenheit gerät. Ein modernes Konzept hat sie allemal aufzuweisen. Sie ist gleichzeitig Betriebssystem und Entwicklungsumgebung und arbeitet mit einem virtuellen Prozessor.

Seite 52

Deutsches mit CPC unter CP/M

So richtig zufrieden kann man als Textverarbeiter nur sein, wenn Umlaute auf dem Monitor nicht als eckige und geschweifte Klammern erscheinen. Wie Sie beim CPC 464 unter CP/M Deutsches auf den Schirm bekommen, ohne wertvollen TPA-Platz zu opfern, lesen Sie auf

Seite 60

Inhalt

Horizont-Erweiterung

Wer mit Computern in fernen Ländern Verbindung aufnehmen will, um sich unerschöpfliche Informationsquellen zu erschließen, braucht nebst Telefon ein Post-Modem. Auf dem c't-Prüfstand mußte sich die galvanisch gekoppelte Modem-Box der Firma Elsa bewähren.

Seite 28

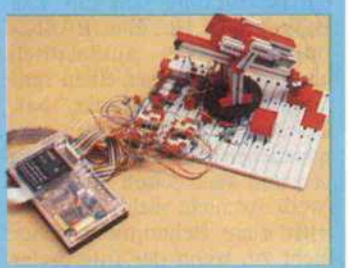
Mehr Platz auf CP/M-Disketten

Die blockorientierte Diskettenverwaltung von CP/M hat den Nachteil, daß immer ein gewisser Teil der Plattenkapazität ungenutzt bleibt. Dies erhöht leider die Anzahl der benötigten Disketten für Sicherungskopien und Archivierung. Warum soll man das nicht ändern, beispielsweise mit einem Programm in Turbo-Pascal?

Seite 84

Schalten und walten

kann man mit dem Computing Interface von Fischer-Technik.

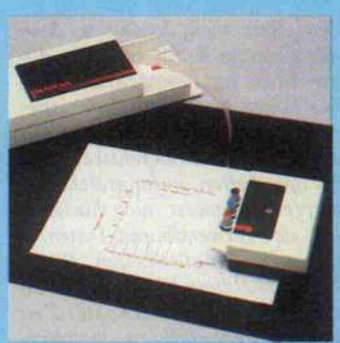


Commodore-Rechner mit User-Port versetzt es in die Lage, Modelleisenbahnen und Roboter zu steuern oder die Redaktions-Kaffeemaschine einzuschalten. Testen Sie mit auf

Seite 42

Der andere Plotter

Ein Plotter muß gar nicht aus einer sperrigen Mechanik bestehen, die den Zeichenstift mit der notwendigen Präzision über das Papier führt. Es geht auch anders. Wir testeten den Penman, in der Hauptsache ein automobiles Malfahrzeug, dessen Bewegungsfreiheit nur durch das Steuerkabel eingeschränkt wird.



Seite 32

Projekte

Die versteckte Uhr
Akkugepufferte Echtzeituhr für jeden Rechner **44**

Schnell, schneller, RAM-Disk
1 MByte Hochgeschwindigkeits-Massenspeicher **76**

Härtetest
c't68ECB-Hardware in der Praxis **117**

Software-Review

Datenanalyse
Neu von Borland: 'Reflex' **34**

Editor für alle Fälle
Programmierbar: 'Pmate' **36**

Turbo-Tutor
Turbo-Pascal-Kursus **122**

PROLOG 64
Interpreter für den C64 **122**

MAXAM
Assembler für CPC 464/664 **124**

Applikation

Bitverdoppler
Ein Bit Speicher - zwei bit Information **113**

Einfach und preiswert
Der serielle Schnittstellenbaustein 8251A **114**

Rubriken

Editorial **3**
Leserbriefe **6**
c't-aktuell **8**
Neue Modelle von Atari **83**
Platinen-Service **43**
Club **120**
Buchkritik **126**
Inserentenverzeichnis **137**
Impressum **138**
Vorschau auf Heft 5/86 **138**
Software-Service **139**

Passendes Buch

(Editorial, c't 2/86)

Es reicht eben nicht, wenn in einer technischen Publikation die Zahlen und die Programmlistings stimmen. Auch der talentierteste Programmierer kann sich mit seinen REM's oder sonstigen Kommentaren zum Quellcode einfach unmöglich machen. Natürlich müssen die Redakteure einer technischen Fachzeitschrift keine Germanisten sein. (Ich bin auch keiner.) Aber sie sollten sich, zumal beim 'Editorial' den Gebrauch solch zweifelhafter Wortschöpfungen wie 'Spitzfinder' und 'hinterfragen' verkneifen. Aber die Versuchung ist groß, ich versteh's. Wer Code entwickelt, neigt dazu, auch Worte zu 'entwickeln', ohne großes Nachdenken. Hier liegt die Verantwortung des Redakteurs, und der Deutschlehrer des Herrn Grell sollte diesen beim nächsten Mal ruhig etwas fester klopfen, unterhalb der Schulter. Das Poster aber ist gut, und daß Sie es gelegentlich anschauen auch. Ich hätte dazu noch die passende Buchempfehlung für Sie, ganz im Sinne des oben Gesagten: 'Die Leiden der jungen Wörter', ISBN 3 7608 0357 1.

Jobst Kilian, Völksen

Wer irrt?

(Einsteigen in CP/M, c't 7/85 ff.)

Laut den CP/M-Büchern von Plate und Pol müßte die BIOS-Routine LISTST den Wert 255 liefern, wenn der Drucker Daten übernehmen kann, und 0, wenn der Drucker nicht bereit ist. Bei meinem 56k-CP/M-2.2/3.01 TA alphaTronic PC werden genau die umgekehrten Werte geliefert. Also hat sich entweder Pol in diesem Punkt geirrt und Plate von ihm abgeschrieben, oder die Routine LISTST ist im BIOS des alphaTronic PC falsch implementiert. Wo liegt der Fehler, bei Plate und Pol oder bei der Triumph-Adler-AG?

Noch eine Frage: Ist die Datensicherheit auf dem Atari 520 ST eigentlich geringer als auf den wesentlich teureren PCs IBM und Kompatible?

Stefan Reeder, Nürnberg

Der Irrtum liegt bei TA. Daß er bislang nicht aufgefallen ist, hat wohl den Grund, daß die LISTST-Routine nur eine Beigabe ist. Das BDOS benutzt sie nicht, und die Programme, die die LIST-Ausgabe direkt übers BIOS abwickeln, sind extrem selten (Beispiel: DESPOOL von Digital Research).

Zu Ihrer zweiten Frage: Hersteller- und modellspezifische Unterschiede zwischen verschiedenen PCs treten nach unserer Erfahrung in der Praxis nicht auf.

(Red.)

Schon lange mit OS9

(Schnelles, schlankes Multiuser-System, c't 1/86)

Zuerst möchte ich Ihnen zu Ihrer Zeitschrift gratulieren, die ich immer mit viel Freude lese. Vor allem die Programme zur Diskettenreparatur unter Dragon-DOS in der c't 2/86 sind wirklich sehr nützlich. Da Ihnen der Dragon anscheinend nicht unbekannt ist, habe ich mich gewundert, daß Sie in Ihrer letzten Ausgabe (c't 1/86) bei den Artikeln über OS-9 und über die Sprache C nicht erwähnt haben, daß das OS-9 schon lange auf dem Dragon 64 läuft, da er den 6809-E-Prozessor besitzt. Des weiteren kann man unter OS-9 auf dem Dragon problemlos in der Sprache C programmieren. Meines Wissens ist der Dragon der einzige relativ preisgünstige Homecomputer, der solche Möglichkeiten zuläßt. Deshalb hätte ich das schon für erwähnenswert gehalten.

Alexander Tubetzkoy, Bad Honnef

Mühsames Patchen

(Rasanter Zugriff, c't 8/85)

Ich habe Ihr Projekt c't86 von Anfang an mitgebaut. Meine letzte Aktion war die Inbetriebnahme der IFC-Unit unter CP/M 2.2. Eine schöne Sache! Nur die Diskettenformate stören mich gewaltig. Ich habe mir deshalb im letzten Jahr bereits das Source Listing des IFC-EPROMs bei Ihnen bestellt. Aber das Patchen ist doch eine mühsame und nicht immer erfolgreiche Angelegenheit. Deshalb meine Frage: Können Sie mir das Source Listing nicht auch auf Diskette schicken? Natürlich gegen Erstattung der Unkosten.

Meine zweite Frage ist: Was muß ich anstellen, um eine mit dem IFC-Monitor formatierte PCDOS-Diskette mit dem Betriebssystem zu beschreiben? Bei Aufruf des SYS-Befehls erhalte ich nur die Meldung 'Kein Platz auf der Diskette'.

Franz Bigge, Darmstadt

Es ist unterdessen ein neues IFC-EPROM entwickelt worden, das verschiedene Verbesserungen aufweist. Die neue Version ist auch als Source Listing auf Diskette erhältlich, mit Patch-Anleitung für andere Diskettenformate. Mit der neuen Software funktioniert auch der SYS-Befehl unter PCDOS.

(Red.)

Immer gleich

Heute möchte ich endlich meine Meinung zu einer oft publizierten Behauptung kundtun. Die Behauptung ist: Ein BASIC-Interpreter ist automatisch langsamer, wenn er einen umfangreichen Befehlssatz hat. Gelesen habe ich dies in verschiedenen Computermagazinen und auch schon mal in c't (weiß nur nicht mehr wo). Nun trifft diese Behauptung sicher nicht zu, wenn der Interpreter nach folgendem Prinzip arbeitet: Die BASIC-Schlüsselwörter sind als 'Token' im Speicher kodiert (dies ist wohl immer der Fall). Wenn der Interpreter bei der Abarbeitung des Programms auf ein solches Token trifft, benutzt er den Wert des Tokens (mit einem Faktor multipliziert) als Index in eine Tabelle, die die Adresse der zugehörigen Routinen enthält.

Nach dieser Methode ist die Zeit, die bis zur Ausführung eines BASIC-Befehls vergeht, immer gleich lang – egal, wie viele Befehle existieren.

Harald Thienel, Waltenhofen

Ihr Argument trifft zu, wenn die Anzahl der BASIC-Schlüsselwörter 128 nicht überschreitet; mehr Befehle kann man mit einem Byte nicht kodieren, denn das gesetzte 8. Bit ist ja im allgemeinen zur Kennzeichnung eines Tokens reserviert. Es existieren aber bereits Interpreter mit einem größeren Befehlssatz. Im übrigen bieten umfangreiche Interpreter zumeist auch hinsichtlich der verwendbaren Datentypen mehr Möglichkeiten. Deren Unterscheidung bei der Bearbeitung des Programms kostet Zeit und führt ebenfalls zu einer Verlangsamung.

(Red.)

Sehr angetan

(c't 68000, c't 10/85 ff.)

Sehr angetan war ich von Ihrem Plexiglas-68000 (was wohl die Post dazu zu sagen hätte?) und möchte Sie daher dazu anregen, nicht immer nur die komplexesten Projekte vorzustellen, sondern auch der Rückmeldung der Nachbauer derselben in Ihrer Zeitschrift Raum zu gewähren. Dabei ist außer Hard- und Software eben auch der mechanische Aufbau interessant – das Auge 'ißt' schließlich mit. In Ihren Tests von Industriegeräten spielt dieser ja auch eine Rolle.

Ich selbst hatte leider zum Austüfteln der 'Mechanik' sehr viel Zeit, da die angekündigte Lieferfrist von 6 um 20 Wochen überschritten wurde. Auch dazu – Lieferbarkeit von Bausätzen und Komponenten von seiten Dritter – wäre gelegentlich ein erklärendes Wort angebracht. Ansonsten (genug gemeckert!) gefällt mir Ihre Zeitschrift recht gut, ragt sie fachlich doch wohltuend aus ihrem Markt heraus.

Walther Mathieu, Aachen

Wir würden gern mehr 'Rückmeldungen' von allgemeinem Interesse veröffentlichen, aber leider erhalten wir fast ausschließlich Anfragen zu Einzelproblemen. In diesem Zusammenhang sei nochmals an die Möglichkeit erinnert, über die 'Club'-Rubrik kostenlos Kontakte herzustellen und Software zu c't-Projekten anzubieten. Zur Lieferfähigkeit Dritter möchten wir uns nicht äußern. Und was den Plexiglas-68000 betrifft, so handelt es sich um ein 'Show-Stück' für Messen und Ausstellungen, dessen Gehäuse wir ausdrücklich nicht zum Nachbau empfehlen.

(Red.)

Aufrüsten problemlos

('Memory full' ade! c't 1/86)

Ich besitze den CPC 464 mit einer Vortex-Speichererweiterung SP-64, die ich mir selbst auf 320 KByte erweitert habe. Das Aufrüsten der Vortex-Erweiterungen ist ohne Probleme möglich. Neben den RAM-Bausteinen benötigt man nur noch einen 74HCT373, der in den letzten freien Sockel (neben einem gleichen Baustein) gesteckt werden muß.

Eckhard Hinrichsen, Hamburg

Bequem überspielen

Seit etwa zwei Monaten bin ich stolzer Besitzer eines Sharp PC1500A. Letztens machte mich ein Bekannter auf Super-Tape aufmerksam, das ja auch auf diesem Rechner läuft. Mein Interesse galt auch der Version für den C64, um dann bequem überspielen zu können. So bekam ich auch während meiner Suche das Heft 4/84 in die Hand und habe daraufhin die ganze Serie nachgelesen. Meine Frage: Sind die BASIC-Befehle genauso genormt wie die ASCII-Zeichen, und wenn nicht, wie realisieren Sie dann die Kommunikation zwischen zwei verschiedenen Rechnern? Wie lauten die Codes für die einzelnen Befehle?

Peter-Hendrik Müns, Kiel

Nichts ist so wenig genormt wie Computer. In der Regel stimmen zwischen zwei BASIC-Rechnern weder der Befehlswortschatz noch die interne Verschlüsselung (Tokenisierung) der Befehle überein. SuperTape ermöglicht es aber, Programme und Daten von einem Rechner auf den anderen zu übertragen. Bevor das BASIC-Programm auf dem Fremdrechner läuft, müssen zumindest die Tokens angepaßt werden. Wie sich dies automatisch, per Programm, bewerkstelligen läßt, zeigt der Beitrag 'Programm-Transfer' in c't 1/86 an der Kombination Apple/PC1500. (Red.)

Noch'n Tip

Daß Sie meinen Atari-520-Tip so positiv aufgenommen haben, hat mich sehr gefreut. Hier ein weiterer sehr nützlicher Tip zum ST-BASIC: Durch folgenden Befehl wird die Grafikausgabe um ca. 50% schneller!

POKE SYSTAB+24,1

Leider kann man dann nicht mehr mit Maus oder Tastatur etwas eingeben, also auch das laufende Programm nicht unterbrechen. Der Normalzustand wird mit

POKE SYSTAB+24,0

wiederhergestellt. Ich hoffe, daß viele Leser diesen Tip anwenden können.

Heinrich Tillack, Hildesheim

Kerbe verdreht

(Aufgestockt, c't 11/85)

In Heft 11/85 berichteten Sie über die 10K-Erweiterung für den Sharp PC-1401. Beim genaueren Durchlesen des Berichtes stieß ich auf ein Problem: Während nämlich in Bild 1 die Speicher-ICs in der richtigen Lage (Kerbe nach unten) abgebildet sind, ist in Bild 2 die Verdrahtung der RAMs mit der Kerbe nach oben dargestellt; die Positionierung der ICs ist allerdings die gleiche geblieben. Ich nehme an, daß dies nur ein Versehen ist, möchte aber auf jeden Fall nicht eher den Lötkolben schwingen, bevor ich in diesem Punkt Gewißheit habe.

Wolfgang Luhmann, Hamm

Der Verdrahtungsplan auf Seite 105 ist korrekt. (Red.)

Völlig verlassen

(Compiler für den Atari ST, c't 2/86)

Nach der Lektüre Ihres Artikels 'Compiler für den Atari ST', der sich unter anderem mit dem neuen MODULA-2-Compiler von TDI auseinandersetzt, möchte ich einige Erfahrungen aus Anwendersicht beisteuern, die manchem Leser auch im Hinblick auf eine etwaige Kaufentscheidung nützlich sein dürfte:

Die besonderen Leistungsmerkmale von MODULA-2, nämlich komplexe Programmsysteme zu unterstützen und dem Programmierer größere sprachliche Freiheiten bis zum Abstieg auf Maschinenebene zu erlauben, bergen naturgemäß erhöhte Fehlermöglichkeiten in sich, denen durch geeignete Unterstützung bei der Fehlersuche Rechnung getragen werden sollte. Während der Compiler in der mir vorliegenden Version 2.02a recht detaillierte Compile-Fehlermeldungen erzeugte, sofern er sich nicht vorher, wie öfters geschehen, durch einen Runtime-Error selber lahmlegte, ist man bei Laufzeitfehlern völlig verlassen; öffenet sich doch nur ein Fenster, das eine von 14(!) Fehlerarten signalisiert, ohne die Fehlerstelle auch nur bis auf Modulebene genau zu bestimmen. Man stelle sich die Aussagekraft eines Overflow-Trap-Errors in einem 30 KByte langen Arithmetik-Programm vor!

Bei der Lokalisierung des Fehlers blieb nichts übrig, als auf bewährte Assembler-Praktiken (Breakpoints usw.) zurückzugreifen, was jedoch vom Zeitaufwand her unzumutbar ist. Es bleibt zu hoffen, daß TDI diesen extremen Schwachpunkt seines Systems in Kürze beheben wird.

W. Spinnler, Bubenreuth

Neger im Tunnel

(Apple wird 'mehrsprachig', c't 2/86)

Die Zeichensätze lassen sich gut umschalten. Sie haben jedoch einen großen Nachteil. Bei Verwendung des Programms 'MULTIPLAN' ist auf leeren Feldern kein Cursor mehr zu sehen. Man sieht hier nur noch 'Neger im Tunnel'. Vielleicht wäre es gut, in einer Ihrer nächsten Ausgaben auf diesen Mangel hinzuweisen.

Günther Schulz, Dinslaken

Es handelt sich um einen Nachteil von MULTIPLAN, das statt der normalen Cursorfunktion den inversen Zeichensatz benutzt. Mit dem MULTIPLAN-Installationsprogramm können Sie dies ändern, indem Sie 'Videx without inverse' selektieren. (Red.)

Welcher Verlag?

(Auf kürzestem Weg, c't 2/86)

Auf der Seite 46 Ihres Magazins erwähnen Sie den Titel 'Geographisches Breitenbuch der BRD', ohne auf den Verfasser oder den Verlag hinzuweisen. Ich bitte Sie, den Autor oder den Verlag mitzuteilen, da ich an diesem Buch interessiert bin und es ohne diese Angaben nicht erhalten kann.

Günter Schirm, Kalletal-Kalldorf

Das Buch heißt exakt 'Geographisches Namensbuch Bundesrepublik Deutschland', Frankfurt, 1981, Band 1/2, Verl. d. Inst. Weitere Längen-/Breitenangaben sind auch guten Atlanten zu entnehmen. (Red.)

Nicht perfekt

(Turbo Inliner, c't 2/86)

Zunächst mal vielen Dank für dieses Utility, das den 'sehr bequemen Weg' endlich tatsächlich bequem macht. Aber wie sollte es anders sein: Nobody is perfect! Am abgedruckten Listing stören mich dann doch ein Fehler und mehrere fehlerträchtige 'Schönheitsfehler': Die Routine 'Calculate Address' berechnet Zwei-Byte-Befehle falsch! Damit z.B. LD DE, (Label) auch dann richtig gehandhabt wird, wenn 'Label' nicht als EXT deklariert ist, sondern im Programm definiert wird, sind folgende Änderungen nötig:

```
PROCEDURE Calculate_Adress;
 BEGIN Strg:='$'+Copy(Newline,4,4);
  VAL(Strg,LabelPosition,Result);
  Offset:=LabelPosition-ProgCounter-1;
  IF Switch THEN Offset:=Offset-1;
  Str(Offset,Strg); ....
```

Ein geändertes Listing mit dieser Verbesserung (und einigen anderen) kann gegen Rückporto von mir bezogen werden.

Michael Bührer, Steinstraße 85, 8000 München 80

Ergänzungen + Berichtigungen

Schnelle Speicher

(c't 1/85, 4/85)

Die schnellen RAM-ICs HM3-65161-5 für den Speicheroszilloskop-Zusatz zum C64 (c't 1/85) und den Logikanalysator-Zusatz (c't 4/85) sind jetzt erhältlich bei Fa. RAIL-electronic GmbH, Großer Biergrund 4, 6050 Offenbach/Main.

Auf kürzestem Weg

(c't 2/86)

Die Programmzeile 170 enthält einen Tippfehler (der beim Programmlauf auf dem Atari ST nicht in Erscheinung tritt). Es muß heißen: 170 dim BG(Q). . .

Turbo-Inliner

(c't 2/86, S.64)

In der Prozedur Form_New_Line muß auf Seite 68, linke Spalte das erste Argument in der Copy-Anweisung von

```
Upcaseline:=Copy(Textline,
ExternalPos,Linelength-
ExternalPos+1);
```

nicht 'Textline', sondern 'Upcaseline' heißen.

Außerdem besteht MACRO80 auch bei dem Programm Unsinn (S.66) auf der .Z80-Direktive (siehe PRN-File, erste Zeile), die im Assembler-Quelltext der Schere zum Opfer gefallen ist.

Apple wird mehrsprachig

(c't 2/86, S. 57)

Der Name des Autors wurde versehentlich falsch angegeben. Der Beitrag wurde verfaßt von Jan H. Haul.

EPROM-Bank für Atari ST

(c't 1/86, S.58)

Die Pull-Up-Widerstände R1 . . . R3 haben den Wert 4k7.

Mikrocomputer von IBM mit RISC-Architektur

IBM stellt einen neuen 32-Bit-Mikrocomputer vor, der für technisch-wissenschaftliche Anwendungen sowie für CAD-Einsatz prädestiniert ist. Das System mit der Bezeichung IBM 6150 ist mit einem Prozessor mit RISC-Architektur ausgestattet.

Der 6150-Mikrocomputer hat eine Zykluszeit von 170 Nanosekunden und eine Leistung von 1,6 bis 2,1 MIPS (Mega-Instruktionen pro Sekunde). Die RISC-Architektur (Reduced Instruction Set Computer) ist auf einen Satz von Instruktionen ausgelegt, die jeweils in einem Maschinenzyklus abgearbeitet werden.

Die offene Systemarchitektur des IBM 6150 erlaubt einen auf die Bedürfnisse des jeweiligen Anwenders abgestimmten Systemausbau. So kann der Zentralspeicher bis auf 4 MByte, die Plattenkapazität auf 210 MByte und die Diskettenkapazität bis auf 2,4 MByte ausgebaut werden. Für die Ein-/Ausgabe enthält der IBM 6150 den 16-Bit-I/O-Bus vom IBM PC AT, wodurch das neue System die PC-Peripherie nutzen kann. Mit einem zusätzlichen PC AT-Coprozessor wird der 6150 kompatibel zum IBM PC AT.

Das Betriebssystem AIX (Advanced Interactive Executive) basiert auf UNIX System V. AIX ist ein Multitasking-/Multiuser-Betriebssystem, das gegenüber UNIX durch eine menügesteuerte Benutzeroberfläche und ein hardwareunterstütztes virtuelles Maschinenkonzept erweitert ist. Außerdem unterstützt das IBM AIX die relationale Datenbank SQL sowie die Programmiersprachen Fortran, Pascal, BASIC und Assembler.

Für die grafische Ausgabe kann an den 6150-Computer das Bildschirmsystem IBM 5080 angeschlossen werden, wodurch der Anwender ein leistungsfähiges CAD-System erhält. Die Palette der für den IBM 6150 angebotenen Monitore reicht vom 12"-Monochrom-Gerät bis zum 15"-Grafikbildschirm mit 1024 × 768 Bildpunkten. Für die Kommunikation ist neben parallelen und seriellen Schnittstellen-Adaptern auch ein PC-Netzwerk-Adapter verfügbar.

Der Kaufpreis des Systems, das im vierten Quartal dieses Jahres ausgeliefert werden soll, liegt zwischen 50 000 und 200 000 DM (jeweils zuzüglich Mehrwertsteuer), je nach Ausbaustufe und Modell.

IBM Deutschland GmbH, Postfach 80 08 80, 7000 Stuttgart 80, 07 11/78 5-0.

VDI lädt zum Softwarekongreß

Vom 13.-15. Mai wird auf dem Hamburger Messegelände von der VDI (Vereinigung der Deutschen Software-Hersteller e.V.) der erste europäische Softwarekongreß 'Euro Software'86' stattfinden. Neben Fachvorträgen über aktuelle Trends und Entwicklungen im Software-Bereich steht für den 14. Mai ein Fachsymposium zum Thema 'Software und Recht' auf dem Programm.

Network GmbH, Wilhelm-Suhr-Str. 14, 3055 Hagenburg, 0 50 33/70 57

Osborne 6 mit 40 MB

Der AT-kompatible Osborne 6 ist jetzt auch in einer 40-MB-Version im Handel. Zur Standardausrüstung gehören ein Bildschirm mit 14 Zoll (dreh- und schwenkbar), 1,2-MB-Laufwerk, 40-MB-Festplatte, deutsche DIN-Tastatur und deutsches Handbuch. Zum Preis von 12 450 DM

sind im Lieferumfang außerdem eine Hercules-kompatible Grafikkarte (720*348), 1 MB RAM auf der Platine, zwei Centronics- und eine serielle Schnittstelle enthalten.

Osborne Vertriebsunion, Dingolfinger Str. 6. 8000 München 80, 0 89/49 10 01

Neuer PC von Multitech

Der neue 'superschnelle' PC 700 PLUS von Multitech ist mit dem Intel 8088-2 ausgestattet und arbeitet mit einer Taktfrequenz von 8 MHz. Der 4,77-MHz-Takt ist hard- und softwaremäßig einschaltbar. Außerdem sind jetzt auch sechs

IBM-kompatible Steckplätze vorhanden. In der Standardausrüstung bietet der 700 PLUS die Speicherkapazität von 256 KB RAM, 8 KB ROM (erweiterbar auf 48 K), zwei Laufwerke mit 360 KB, die Farbgrafik-Karte mit BAS und RGB-TTL sowie eine parallele Centronics- und eine RS-232-Schnittstelle. Für den Preis von circa 5050 DM werden ein Monochrom-Monitor mit allen nötigen Verbindungskabeln, die Betriebssysteme MS-DOS 2.11 und Concurrent CP/M-8 sowie ein Textverarbeitungsprogramm mitgeliefert. Den Vertrieb übernimmt die Firma Ce-Tec. Andere Ausstattungskombinationen,wie zum Beispiel 640 KB RAM, Harddisk mit 10 oder 20 MB und Farbmonitor, sind ebenfalls möglich.

Ce-Tec, Kornkamp 4, 2070 Ahrensburg, 0 41 02/49 01-0

Maschinennah in Turbo-Pascal

Maschinennahes Programmieren unter Turbo-Pascal ermöglicht das Software-Routinen-Paket 'Turbo Machine' von Lauer & Wallwitz. Es enthält unter anderem Interrupt- und BIOS-Service, einen Menügenerator, Unterstützung von DOS 2.x und 3.x, Tastatur-Redefinition (Makros), serielle Kommunikation sowie weitere Prozeduren und Funktionen, die den Turbo-Pascal-Programmierer erfreuen. 'Turbo Machine' ist mit deutschsprachiger Dokumentation und in spezifischer Rechneranpassung für Sirius, Wang, IBM PC/XT/AT und Kompatible erhältlich.

Lauer & Wallwitz, Gesellschaft für angewandte Programmierung, Erlkönigweg 9, 6200 Wiesbaden, 0 61 21/42 771

SYNELEC
DIE PERIPHERIE-PROFIS

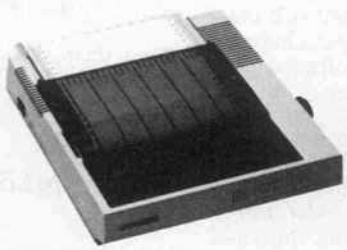
Präsentieren:

Hannover Messe '86
CeBIT —— 12.-19. 3. 1986
Halle 17, Stand 403/500

MSP 25

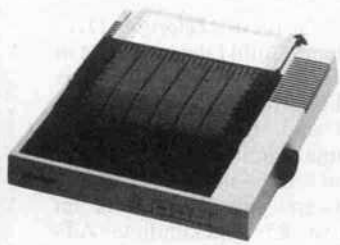
MSP 15

CITIZEN



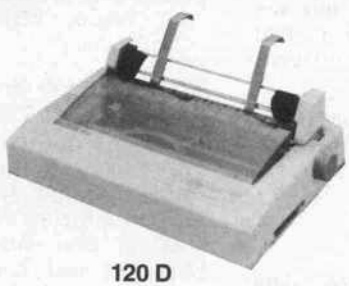
MSP 20

Zwei Jahre ★ Vollgarantie



MSP 10

Premiere 35



120 D

Matrixdrucker

120 D
- ☐ 120 Zeichen pro Sek.
- ☐ 25 Zeichen pro Sek. (NLQ)
- ☐ 80 Zeichen pro Zeile bei 10 CPI

MSP 10/15
- ☐ 160 Zeichen pro Sek.
- ☐ 40 Zeichen pro Sek. (NLQ)
- ☐ 80/136 Zeichen pro Zeile bei 10 CPI

MSP 20/25
- ☐ 200 Zeichen pro Sek.
- ☐ 50 Zeichen pro Sek. (NLQ)
- ☐ 80/136 Zeichen pro Zeile bei 10 CPI

Alle Citizen-Matrixdrucker sind IBM® PC- und Epson®-kompatibel

Typenraddrucker

Premiere 35
- ☐ 35 Zeichen pro Sek.
- ☐ Proportionalschrift
- ☐ 8 KByte Puffer
- ☐ Standard USASCII Zeichensatz

Rufen Sie uns an! **089/51 79-0**

c't 4/86

Info-Coupon

SYNELEC DATENSYSTEME GmbH

Postfach 151727 · D-8000 München 15 · Tel. 089/5179-0 · Telex 5212289 syn d
Geschäftsstelle Frankfurt: Tel. 06081/8061 · Telex 415339 oehlm d
Geschäftsstelle Düsseldorf: Tel. 0211/350236 · Telex 8588914 syn d

☐ Ich bitte um Informationen über Citizen-Drucker ☐ Ich bin Händler

Name _____
Firma _____
Abt. _____
Straße _____
Ort _____
Telefon _____

KSp 1234

Spezialanwendungen von Taylorix

Der EDV-Anbieter Taylorix will sich verstärkt standardisierten Anwendungen in Klein- und Mittelbetrieben verschiedenster Unternehmenszweige widmen. Im Mittelpunkt stehen Branchenlösungen auf Mikrocomputern. Für die Umstellung auf EDV bietet Taylorix das Erfassungssystem BEG an, das mit den entsprechenden Programmen zum Komplettpreis von 3000 DM geliefert wird. Die Verarbeitung der Betriebsdaten erfolgt dann in den Taylorix-Rechenzentren. Für PC-Anwender (MS-DOS) stehen die Service-Rechenzentren über DFÜ-Anschluß zur Verfügung. Taylorix ist in Halle 1/Stand 6f2/6gl zu finden.

Pressestelle der Taylorix Organisation, Postfach 8829, 7000 Stuttgart 1, 07 11/82 09-0

VIENNA erfährt Neuigkeiten

Die Northern Telecom Data Systems GmbH demonstriert in Halle 1/Stand 4C1/4C3 eine Reihe von System-Verbesserungen und Neuheiten für die Vienna-Familie. Dazu zählt vor allem die neue Netzwerk-Software und der bereits im Herbst 85 angekündigte AT-kompatible Rechner mit weißem, hochauflösenden Bildschirm und 640 KByte Arbeitsspeicher.

Northern Telecom Data GmbH, Lyoner Str. 30, 6000 Frankfurt 71, 069/6 69 70

Neuentwickelter Akustikkoppler

Der neue Akustikkoppler 'dataphon s 21/23 Kombi' hat einen eingebauten Geschwindigkeitswandler, der bei Bedarf automatisch die Umschaltung von 1200 auf 75 Baud für den Rückkanal vornimmt. Das Gerät von Woerltronic ist in CMOS-Technologie konstruiert, ermöglicht induktive oder akustische Kopplungen und verfügt über BTX-Fähigkeiten. Der 'dataphon s 21/23 Kombi' wird ab Mitte April zum Preis von ca. 400 DM erhältlich sein, kann aber schon in Halle 18 EG./Stand 103 bewundert werden. Der Hersteller machte keine Angaben über die postalische Zulassung.

Wörlein GmbH & CO. KG., Hindenburgstr. 37, Postfach 4, 8501 Cadolzburg, 0 91 03/82 94

Video und Computer von Sony

Zur CeBIT-Messe präsentiert Sony in Halle 6 auf rund 1000 m² Ausstellungsfläche seine neuen Produkte für die Bild- und Bürokommunikation. Dazu gehören ein optoelektronisch arbeitendes Büroinformationssystem mit beschreibbarer Laserplatte und 3,2 Gigabyte Speicherkapazität, das Textverarbeitungsprogramm M 10., das mit dem ebenfalls neu vorgestellten 'Manager Computer' SMC-210 zusammenarbeitet, und das 'VIEW-System', das die höchste Form des interaktiven Video repräsentieren soll. Im Videokonferenzstudio wird neben 'klassischen Stand-alone-Lösungen' auch zu sehen sein, wie Bildschirmtext in interaktive Video-Systeme integriert werden kann. Außerdem werden interaktive U-matic- und Bildplatten-Lösungen vorgestellt.

Sony Deutschland GmbH, Public Relations, Hugo-Eckener-Straße 20, 5000 Köln 30, 02 21/59 66-379-381

Supercomputer aus deutschen Landen

Ein völlig neues Konzept im Bau von ultraschnellen Rechnern verfolgt die junge Computerfirma iP-Systems mit dem Modell TX2. Nicht spezielle, hochgezüchtete Hardware ist das Geheimnis dieser neuen Maschine, sondern die Synthese aus Hardware-Architektur und einer neu entwickelten Software-Methodik. Da nur marktübliche Elektronikbauteile eingesetzt werden, ermöglicht dieses Konzept zudem hohe Rechnerleistung zu einem Bruchteil des Preises herkömmlicher Großrechner.

Die Hardware des TX2 besteht aus einer Vielzahl völlig gleichartiger Module, die zu einer baumartigen Struktur verknüpft sind. Jedes Modul basiert auf einer 80286-CPU, verfügt über einen eigenen Speicher und kommuniziert mit seinen Nachbarmodulen über eine 16-Bit-Parallel-Schnittstelle mit 4 bis 8 MByte/s. Die hohe Rechnerleistung entsteht nun dadurch, daß alle Module parallel an Teilaufgaben eines Problems arbeiten.

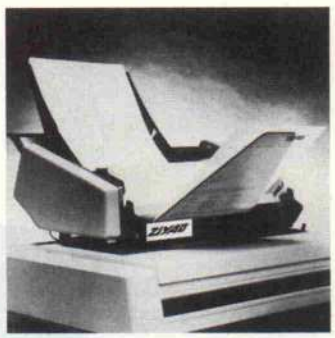
Die baumartige Struktur des TX2 ist speziell für neue Parallel-Algorithmen ausgelegt, die an der Universität Karlsruhe entwickelt wurden. Sie erlauben es, den Geschwindigkeitsvorteil der Parallelität für mehr Anwendungsbereiche zu nutzen, als dies bei anderen Parallelrechnern möglich ist. Geeignete Aufgaben für den TX2 sind beispielsweise komplexe Simulationen und Optimierungen sowie Datenbanken und Expertensysteme. Außerdem läßt sich die Rechengeschwindigkeit des Computers durch Hinzufügen oder Entfernen von Modulen verändern, ohne daß sich die übrigen Systemeigenschaften ändern oder die Software angepaßt werden muß.

Zur Erstellung von Programmen eignen sich die 'parallelen' Erweiterungen von Pascal, Modula-2, C oder Fortran. Die Umsetzung vorhandener Programme kann mit Precompilern zum automatischen Parallelisieren geschehen. Bei der Sprache Prolog und bei Datenbank-Software wird der Programmierer überhaupt nicht mehr mit dem Problem der Parallelität konfrontiert.

Es scheint tatsächlich so, daß der technische Fortschritt sich diesmal nicht in den Schmelztiegeln der Chip-Hersteller kristallisiert hat, sondern in den Studierstuben der mathematischen Grundlagenforschung. Der Firmengründer war selbst einer der maßgeblichen Forscher auf dem Gebiet neuer Algorithmentypen. Und iP-Systems gibt die Leistung des TX2 mit 4 Giga-Instruktionen pro Sekunde an, bei optimaler Auslastung der Hardware. Der Preis für das System soll so um die 200 000 DM liegen. Erste Modelle hofft man etwa in einem Jahr ausliefern zu können, erste Informationen erhält man auf der CeBIT in Halle 17, Stand 204.

Dreifache Papierzuführung von Ziyad

Die Dreifach-Papierzuführung 'Z-150 A' von Ziyad holt Briefbögen und Umschlag aus einer der drei Vorratskassetten und führt sie automatisch dem Drucker zu. Die Ablage erfolgt



in der richtigen Reihenfolge und erledigt die Zuordnung der Umschläge im gleichen Arbeitsgang. Das Gerät (Kostenpunkt 1750 DM) wird in Halle 18/Stand 2104/2203 vorgeführt.

Ziyad (Europe) GmbH, Teilfeld 5, 2000 Hamburg 11, 0 40/36 29 23

Integrierte Lösungen bei IBM

Die Schwerpunkte des Ausstellungsprogramms der IBM Deutschland sind integrierte Lösungen auf den Gebieten Bürokommunikation, CIM und Telekommunikation. Der IBM PC als intelligente Dialogstation oder selbständiges System spielt bei vielen Anwendungen eine wesentliche Rolle. Neben zahlreichen Branchenprogrammen für das Handwerk werden unter anderem Beispiele der grafischen Datenverarbeitung mit der Programmserie ASSISTANT, Textverarbeitung mit Text3 sowie verschiedenartige Kombinationen des IBM PC mit dem Telefon und der Einsatz eines PC als Fernschreiber oder Teletex-Gerät demonstriert. Auf der Dachebene der CeBIT-Halle gibt es eine integrierte CIM-Lösung, Möglichkeiten der professionellen Anwendungsentwicklung, Datenbankabfragen über internationale Netze und BTX-Anwendungen zu sehen. Außerdem präsentiert die IBM ihre OSI-Produkte.

IBM Deutschland GmbH, Presseabteilung, Postfach 80 08 80, 7000 Stuttgart 80, 07 11/785-0

MEGA ATARI

Noch mehr Speicherkapazität, um noch mehr zu leisten. Der neue ATARI 520 ST+ hat dieses große Plus für professionelle Computeranwendung. 1 MB steht im Arbeitsspeicher zur Verfügung. Somit können jetzt komplexe Operationen schneller durchgeführt und anspruchvollste Anwenderprogramme mit extrem hohem Speicherbedarf bearbeitet werden.

Den professionellen Anwendern spart der neue ATARI 520 ST+ damit Zeit und Geld. Denn. ATARI bietet den neuen „MEGA-ATARI" zu einem ungewöhnlich niedrigen Preis an. Viele Softwareprogramme werden für den neuen ATARI 520 ST+ bereits angeboten.

Fragen Sie danach beim Fachhandel mit dem blauen ATARI-Schild an der Tür.

Neu. ATARI 520 ST+

1 MB RAM

ATARI 520 ST+ incl. Monitor SM 124, Floppy SF 354.

DM 2.998,-

unverbindliche Preisempfehlung

ATARI®

. . . wir machen Spitzentechnologie preiswert.

Ende des Kabelchaos

Digitale Steuerung für Modellbahnen

Zugreisende lebten noch bis vor wenigen Jahren unter dem Damoklesschwert verheerender Massenkarambolagen, heraufbeschworen durch mangelhaft gesicherte Strecken und allgemeine Konfusion im Weichenstellwerk. Dies betraf natürlich nicht die Passagiere der Deutschen Bundesbahn, sondern die 'gedachten' Insassen des heimischen Modell-Intercity auf der Strecke Keller–Waschküche und zurück; planmäßige Abfahrtszeit: nach Feierabend. Ein großer Modellbahn-Hersteller will dem Chaos jetzt ein Ende machen, natürlich mit Computer-Hilfe.

Bislang hatte der gut ausgerüstete Hobby-Fahrdienstleiter Stellwerke vom Format einer Tischtennisplatte zu bedienen, die auf der Vorderseite mit einer Unzahl von Drehschaltern oder Schleifkontakten ausgerüstet waren und über ein undurchdringliches Kabelgewirr auf der Rückseite die Verbindung zu den Signalen, Weichen, Drehscheiben und Schranken herstellten. Immer häufiger wurde menschliches Versagen zur Ursache dafür, daß der eigentlich für die Einfahrt in den Bahnhof vorgesehene Personen-Triebwagen vorzeitig überraschend falsch abbog und in den erdnußbeladenen Güterzug auf dem Nachbargleis donnerte.

Diese kaputte heile Welt wird systematisch zerstört, seit Märklin 1984 eine digitale Modellbahnsteuerung präsentierte, die ab 1985 auch an Homecomputer angeschlossen werden konnte. Doch damit nicht genug: 1986 wird das Ganze auch noch um das elektronische Fahrstraßen-Stellpult 'memory' erweitert. Es ermöglicht den gleichzeitigen Betrieb von 80 Zügen bei 256 Weichen auf bis zu 24 gespeicherten Fahrstrecken. Sämtliche Funktionen von rollendem und stehendem Material sind dabei auf bloßen Knopfdruck schalt-und kombinierbar. Das gesamte System läßt sich mit Hilfe einer Zusatzerweiterung von einem Homecomputer steuern. Märklin will 'memory' allerdings nur als ersten Schritt auf dem Weg zu einer universellen Steuerung für alle Modellbahnsysteme und Spurweiten gewertet wissen.

Weil bis heute ein unabhängiger Betrieb mehrerer Züge und Lokomotiven nur auf einer Anlage in Dreileiter-Technik möglich ist, tüftelt man zur Zeit an speziellen Dekodern für die Loks der Zweileiter-Freaks, damit auch sie so schnell wie möglich den Anschluß ans digitale Gleis finden. Und wenn's soweit ist, herrscht endlich Ordnung auf den Schienen deutscher Modellbahner, denn wo erstmal die Software stimmt, da irrt sich so ein Computer selten.

Brother Vertriebspartner. Einer ist in Ihrer Nähe.

1000 Berlin: Computer-Shop Minhoff, 030/86 06 42. **1000 Berlin 10:** Cosinus Elektronik, 030/31 06 66. **1000 Berlin 31:** Computer-Shop Minhoff, 030/854 50 55. **1000 Berlin 44:** anyware, 030/623 67 11. **1000 Berlin 62:** Data Systems, 030/784 90 16. **2000 Hamburg 1:** Büro-Dienst Tietz & Scheuren, 040/33 18 51. **2000 Hamburg 13:** Holtkötter, 040/45 79 54. **2000 Hamburg 36:** City Comp, 040/35 31 31. **2000 Hamburg 70:** Holtkötter, 040/66 98 10. **2800 Bremen:** ComputerLand Bremen, 0421/1 50 51. **3000 Hannover:** Frings & Kuschnerus, 0511/32 66 41. **3100 Celle:** Philipp Wolff, 05141/207–0. **3400 Göttingen-Weende:** BEZ Wiederholdt GmbH & Co., 0551/3 40 31. **3500 Kassel:** Hermann Fischer OHG, 0561/70 00 00. **4000 Düsseldorf:** B & B GmbH, 0211/37 10 51–52. **4000 Düsseldorf:** Helmut Rennen GmbH & Co. KG, 0211/30 60 98. **4000 Düsseldorf 1:** GVR Datensysteme GmbH, 0211/68 00 90. **4000 Düsseldorf 11:** Plankopie Heerdter, 0211/5 89 081. **4020 Mettmann:** GCS Ges. für Computerisierung, 02104/7812. **4050 Mönchengladbach:** Data Team, 02161/8 61 46. **4100 Duisburg 1:** N S E Datensysteme, 0203/66 60 91. **4190 Kleve-Kellen:** Feldmann & Luft, 02821/95 66. **4200 Oberhausen 1:** Hobert Computing GmbH, 0208/2 00 51. **4300 Essen 1:** ADT Datentechnik GmbH, 0201/77 38 78. **4300 Essen 1:** Computass GmbH, 0201/23 35 55. **4300 Essen 1:** Winkler & Müller OHG, 0201/23 33 95. **4350 Recklinghausen:** Computer-Zentrale, 02361/4 57 08. **4400 Münster:** Dr. Ehm Mikrocomputer, 0251/79 38 56. **4600 Dortmund 1:** Büro-Studio Bolz, 0231/52 77 13–16. **4600 Dortmund 1:** Grüninger & Co. KG, 0231/52 85 08. **4630 Bochum:** bodata Systemhaus GmbH, 0234/4 36 77. **4630 Bochum:** Micro-Computer Post GmbH, 0234/4 39 03. **4770 Soest:** J. Dahlhoff, 02921/1 25 82. **4800 Bielefeld 1:** Elektronisches Rechenzentrum, 0521/2 50 35. **4800 Bielefeld 1:** PDS Computer GmbH, 0521/2 40 24. **4800 Bielefeld 14:** Microtec GmbH, 0521/4 55 81. **5000 Köln:** Büromaschinen Braun, 0221/21 91 71. **5000 Köln 1:** ComputerLand Köln, 0221/21 96 67. **5000 Köln 50:** Plankopie, 0221/3 90 30. **5060 Bergisch-Gladbach 2:** KBJ Data-Systems, 02202/3 88 84. **5100 Aachen:** Bürotronic Müller, 0241/3 46 51. **5276 Wiehl/Bomig:** R. Halstenbach Software-Service, 02261/7 20 51. **5300 Bonn 1:** B.T.S. Vertriebsges. mbH, 0228/65 10 60. **5500 Trier:** Lehr Bürocenter GmbH, 0651/2 50 44. **5600 Wuppertal 1:** Plankopie, 0202/30 70 51. **5600 Wuppertal 2:** Meiers Büroorganisation, 0202/55 60 60. **5600 Wuppertal 11:** UNA-Computertechnik, 0202/73 02 61. **5650 Solingen 11:** G. Dittrich, 0212/33 10 44. **5860 Iserlohn:** Möller Bürotechnik GmbH, 02371/4 00 22. **6000 Frankfurt:** ComputerLand Vertriebsges. mbH, 069/23 01 41. **6000 Frankfurt:** Computer-Systems Sander + Kositzky, 069/65 43 43. **6000 Frankfurt:** GES Computer, 069/44 77 66. **6000 Frankfurt 1:** bco Büro-Computer + Organisations GmbH, 069/55 04 56. **6072 Dreieich:** IMA GmbH, 06103/6 72 02. **6090 Rüsselsheim:** Delta Computershop, 06142/4 62 66. **6103 Griesheim:** Lutz Büro- und Datentechnik GmbH, 06155/6 10 01. **6108 Weiterstadt 2:** EDV-Studio Fischer, 06150/5 11 56. **6113 Babenhausen 4:** Seipel Consulting GmbH, 06073/39 84. **6200 Wiesbaden:** Natic + EDV, 06121/52 10 91. **6250 Limburg:** wirbürosysteme, 06431/2 60 88. **6330 Wetzlar:** Wilfried Beck, 06441/4 50 26. **6600 Saarbrücken:** Willkomm Computer- und Textsysteme GmbH, 0681/49 80 11. **6733 Haßloch/Pfalz:** Reimer Büromaschinen, 06324/10 01. **6740 Landau/Pfalz:** Data Service, 06341/8 30 72. **6750 Kaiserslautern:** Data Service, 0631/1 60 81. **6800 Mannheim 24:** Gauch & Sturm GmbH, 0621/8 50 04. **6800 Mannheim 31:** DSV Industrie- und Bürosysteme GmbH, 0621/7 26 31. **6800 Mannheim 31:** Ingo Lächele Electronic Vertrieb, 0621/72 22 75. **6905 Schriesheim:** Nassauer, 06203/67 34. **7000 Stuttgart 1:** ComputerLand Stuttgart, 0711/29 44 18. **7000 Stuttgart 1:** Kübler Büro- und Datentechnik, 0711/61 06 51. **7000 Stuttgart 30:** Ramsauer, 0711/85 07 75. **7120 Bietigheim-Bissingen:** M C A, 07142/5 10 99. **7500 Karlsruhe:** C D S, 0721/81 10 35. **7500 Karlsruhe:** Fischer Daten-Systeme, 0721/17 20. **7500 Karlsruhe 1:** Data Service, 0721/35 59 57. **7730 Villingen-Schwenningen:** Benischek, 07721/5 40 65. **7750 Konstanz:** Heimpel, 07531/6 21 07. **7750 Konstanz:** Elektronik Obser, 07531/2 29 29. **7800 Freiburg:** Data Concept, 0761/3 00 49. **7800 Freiburg:** Kühn & Weyh Computervertrieb, 0761/885 02 00. **7800 Freiburg:** Meisterknecht, 0761/3 16 66. **7890 Waldshut-Tiengen:** Heimpel, 07741/30 18. **7900 Ulm:** Interplan TB Software, 0731/2 69 49. **8000 München 2:** Kaut-Bullinger & Co. GmbH, 089/622 62 56. **8000 München 90:** Kaut-Bullinger & Co. GmbH, 089/6 22 60. **8000 München 22:** ComputerLand München, 089/28 60 81. **8000 München 50:** Schulz Bürozentrum GmbH, 089/1 48 21. **8000 München 81:** Bayer & Schäfer GmbH, 089/928 00 40. **8012 Ottobrunn:** PC-Service GmbH, 089/609 62 08. **8013 Gilching:** CPV-GmbH, 08105/14 95. **8032 Gräfelfing:** Pro-Ce GmbH, 089/85 10 43. **8033 Planegg:** Lutz & Farrenkopf GmbH, 089/55 81 91. **8033 Martinsried:** Bax Büromaschinen-Service, 089/85 79 60. **8039 Puchheim-Bahnhof:** Basys GmbH, 089/80 10 69. **8057 Eching:** Lange Computer GmbH, 089/319 10 77. **8070 Ingolstadt:** Herkommer & Gutbrod GmbH & Co., 0841/5 90 71. **8300 Landshut:** Bayer & Schäfer GmbH, 0871/5 13 21. **8473 Pfreimd:** EDV-BV Vertriebs- und Service GmbH, 09606/14 11. **8500 Nürnberg:** HIB GmbH Computerladen, 0911/45 22 11. **8500 Nürnberg 79:** Schuster & Walther GmbH, 0911/28 40 41. **8586 Gefrees:** Integral GmbH, 09254/700. **8600 Bamberg:** A + R Kutz GmbH, 0951/2 78 08. **8900 Augsburg:** Bayer & Schäfer GmbH, 0821/15 70 81–83. **8901 Augsburg-Vogelsang:** Video & Computer GmbH, 0821/48 20 76. **8910 Landsberg:** Lamprecht Büro + Datentechnik GmbH, 08191/4 66 28. **8972 Sonthofen:** Recla & Zschörnig Computersysteme, 08321/8 77 70.

NEUE ZEICHEN SETZEN.

WGJ HH

Mit dem 24-Nadel-Druckkopf vom Brother 2024L+. Plus ladbare Zeichensätze. Plus 14,6K Download-Speicher.

Im wahrsten Sinne des Wortes: Neue Zeichen setzen mit der extrem hohen Auflösung vom Brother 2024L+. Das ergibt die exzellente Schriftqualität – vergleichbar mit dem Typenraddruck – für Dokumentenausdrucke und Textverarbeitung bis hin zu Werbebriefen. Auch die übrige Ausstattung setzt wahrhaft neue Zeichen: Download-Funktion für ladbare Schriften und zum Erstellen individueller Zeichensätze – spielend mit dem Brother Zeichen-Editor. Dazu als Option 10 weitere Schriften. Hochgeschwindigkeit von max. 160 Z/sec. bei Datendruck und auf Tastendruck max. 96 Z/sec. bei Elite-Schönschrift.
Zeichenkapazität pro Zeile: 136 – 272, je nach Schriftart. Kompatibel mit Diablo 630 und Epson LQ-1500 oder IBM. Originalausdruck mit bis zu 4 Kopien. Vier Schriftteilungen incl. Proportional. Fettdruck, Breitschrift, Unterstreichautomatik und vieles mehr. Dazu eingebauter Formulartraktor und interessantes Sonderzubehör wie Einzelblattzuführung.
Schnittstellen Centronics Parallel oder V24 (RS-232C). Übrigens: Der Brother 2024L+ hat die Zulassung als Teletex-Drucker.
Wie schön und schnell der 2024L+ neue Zeichen setzt, davon können Sie sich überzeugen. Nehmen Sie die Einladung wörtlich und gehen Sie zu einem der nebenstehenden Fachhändler.

Einladung

Gegen Vorlage dieser Einladung hat der Überbringer Anspruch auf eine kostenlose und unverbindliche Vorführung mit Probedrucken auf dem Brother 2024L+.
Brother International GmbH, Im Rosengarten 14, 6368 Bad Vilbel.

brother
Die Zukunft heute

BROTHER INDUSTRIES, LTD., Nagoya/Japan

CP/M-Simulator für Atari ST

Omicron bietet für den Atari ST einen CP/M-Simulator an, der gegenüber dem als Freeware verbreiteten einige Vorteile aufzuweisen hat. Laut Herstellerangaben sind die hervorstechenden Merkmale dieses Simulators höhere Geschwindigkeit, ladbarer Zeichensatz, der die Darstellung von Umlauten ermöglicht, sowie einstellbare Steuerzeichen- und Tastatur-Funktionen für die Simulation beliebiger CP/M-Rechner. Ein erweiterter CCP und Download von CP/M-Software anderer Rechner über RS-232-Schnittstelle ist für den Preis von 150 DM im Lieferumfang enthalten.

Omicron-Software, Erlachstr. 15, 7534 Birkenfeld, 0 70 82/53 86

DFÜ mit PC

Das Kommunikationsprogramm CTK PC-DIAL zur Datenfernübertragung für PC/XT/AT arbeitet mit einer Übertragungsgeschwindigkeit von bis zu 9600 Baud. Zusätzlich wird ein Akustikkoppler oder ein Post-Modem benötigt. Das Programm bietet unter anderem automatische Speicherung eingehender Daten auf Diskette und unterstützt Modems mit automatischer Wahlwiederholung. Unter MS-DOS 2.X kann zwischen verschiedenen Dateiverzeichnissen umgeschaltet werden. Die deutsche Version ist mit Handbuch für 454 DM (englische Version 340 DM) erhältlich.

CTK Computer-, Text- und Kommunikationssysteme GmbH, Dolmanstr. 82, 5060 Bergisch Gladbach 11, 0 22 04/63 06 1

Lohnsteuer-Tabelle 86

Für Rechner von IBM, Apple und Commodore gibt es das gesamte Lohnsteuer-Tabellenwerk auf Diskette für einen Preis von 95 DM. Aus der gleichen Tabellenserie stammen auch die 'Einkommensteuer-Tabellen 1975 - 1987', die außerdem auch als Atari-Diskette lieferbar sind.

Haus der Buchhaltung, Postfach 101412, 4100 Duisburg

GEM für PC-D

GEM-Software läuft inzwischen auch auf dem PC-D von Siemens. GEM-Desktop, GEM-Draw, GEM-Collection (GEM-Paint und GEM-Write), GEM-Graph und GEM-Wordchart werden von Digital Research-Distributoren zu Preisen zwischen 179 und 699 DM angeboten. Für die optimale Nutzung der GEM-Software auf dem PC-D empfiehlt Siemens folgende Systemausstattung: PC-D mit 512 KB, Bildschirm-Controller für Grafik, zwei Diskettenlaufwerke (besser noch ein Diskettenlaufwerk und eine Festplatte), eine Maus und den Siemens-Drucker PT88/89.

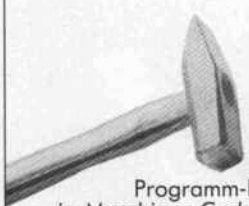
Siemens AG, Zentralstelle für Information, Postfach 103, 8000 München 1, 0 89/28 09 722

PC-Vernetzung auf Ethernet-Basis

Für die MS-DOS-3.1-kompatible Netzwerk-Software NetWare von der Novell Inc. wurde von S&K das leistungsfähige Netzwerk 'SK-NET' entwickelt. Die Netzkontrollplatine mit 68000-CPU und 128 KByte Firmware entlastet den Prozessor des PCs vom zeitraubenden Netzmanagement. Die Karte ist für 3414 DM erhältlich und besitzt einen Transceiver für den Übergang auf Cheaper-Net, einer Low-Cost-Version von Ethernet, die ebenfalls 10 MBit/s überträgt. Ein zweiter Ausgang erlaubt den Anschluß eines externen Ethernet-Transceivers. Theoretisch können mit SK-Net bis zu 1024 PCs vernetzt werden. In der Praxis werden mit bis zu 50/100 PCs kurze Antwortzeiten sowie eine optimale Abwicklung von Multiuser-Anwendungen realisiert. Der Preis für NetWare liegt bei 6840 DM.

Schneider & Koch & Co Datensysteme GmbH, Haid- und Neustr. 7, 7500 Karlsruhe 1, 07 21/6 05 21

Für alle, die in C programmieren wollen: C-Tools



Lattice C-SPRITE Source Debugger Vers. 2.01 MS-DOS

Programm-Debugger der sowohl in Maschinen-Code als auch im Source-Code Level arbeitet. Umfassende Prüfung von C-Programmen über Funktionsnamen, Variablennamen und C-Datentypen oder über Adressen, Befehle, Source-Code Zeilen usw. Macro-Kommandos, Batch-Verarbeitung und Umleitung der Debugger-Ausgaben auf die serielle Schnittstelle gehören ebenso zum Leistungsumfang dieses hervorragenden Softwarewerkzeugs wie eingebaute Hilfstexte. Auch debugging von Overlays ist möglich.

Bestell-Nr. 131010 **DM 718,–**

Lattice Screen Editor Vers. 1.00A MS-DOS

Der Lattice Screen Editor ist ein sehr schneller, anpassungsfähiger und leicht zu erlernender Programm-Editor. LSE ist in Lattice-C geschrieben und wurde schon erfolgreich auf mehrere verschiedene Betriebssysteme übertragen. LSE kann sowohl direkt in den Video-RAM schreiben als auch über BIOS-Call's arbeiten. Das mitgelieferte Installations-Programm ermöglicht die Änderung der Funktionstastenbelegung, der Menüs sowie der Hilfstexte. LSE verfügt über spezielle Eingabe-Modi zur Eingabe von Assembler- oder C-Programmen. Für die Bearbeitung von Text-Dateien ist Wortumbruch möglich.

Bestell-Nr. 131008 **DM 438.–**

PreC „lint"-Utility V.1.34 MS-DOS

Ein dem Programm „lint" entsprechendes Source-Code Prüfsystem. Sie können damit Ihre Programme vor dem compilieren auf Syntax-fehler, Variablen-Typ-Fehler, Parameter-Übergabefehler usw. kontrollieren. Damit finden Sie Fehler, die durch den Compiler nicht erkannt werden und unter Umständen erst nach längerer Laufzeit des Programms oder beim Versuch der Übertragung auf ein anderes System auftreten. Wenn Sie systemunabhängig programmieren wollen, ist PreC ein absolutes Muß.

Bestell-Nr. 131009 **DM 1365.–**

Alle Preise incl. MwSt.
Versand per Nachnahme

Wir informieren Sie gerne. **Wählen Sie: 07841/4500**

PFOTENHAUER Microcomputer-Anwendung
Postfach 1312 · Neulandstraße 16 · 7590 Achern · Telex 752100 msoft

pfotenhauer
MICROCOMPUTER-ANWENDUNGEN

Aktuelle DATA BECKER-Buchhits

Der neue ATARI ist eine Supermaschine! Aber nur der richtige Einstieg garantiert den professionellen Umgang damit. Deshalb sollte dies Ihr erstes Buch sein. Eine Einführung in Handhabung, Einsatz und Programmierung des ST: die Tastatur, die Maus, der Editor, der erste Befehl, das erste Programm, der Anschluß der Geräte u.v.m. Dieses Buch ist ein Muß für jeden Einsteiger!
ATARI ST für Einsteiger, 262 Seiten, DM 29,–

Sie haben den Einstieg auf dem ATARI ST geschafft? Dann werden Sie mit diesem Buch zum Profi. Aus dem Inhalt: Datenfluß- und Programmablaufpläne, Grafik- und Soundprogrammierung, Sortierverfahren, Dateiverwaltung und viele nützliche Tips. Mit einer Befehlsübersicht incl. der nicht bekannten Befehle!
Das große BASIC-Buch zum ATARI ST, 268 Seiten, DM 39,–

Mit diesem Buch wird Ihnen die Erstellung von 3D-Grafiken in Maschinensprache leicht gemacht. Von einer Einführung in Assembler über die nötige Theorie bis zur Grafikanimation in atemberaubender Geschwindigkeit reicht das Spektrum dieses Buches. Außerdem enthält es spezielle Grafikroutinen, die schneller sind als alles bisher dagewesene. Da wird Echtzeitanimation erst möglich.
3D-Grafikprogrammierung zum ATARI ST, ca. 300 Seiten, DM 59,–

Damit Sie die hervorragenden Fähigkeiten Ihres Rechners richtig ausnutzen können, brauchen Sie auch die entsprechende Software. Zeichenprogramme wie GEM-DRAW, DEGAS oder NEOCHROME sprechen für sich. Dieses Buch beinhaltet nicht nur ausführliche und leicht verständliche Handbücher, sondern liefert auch wertvolle Tips und Tricks beim Umgang mit diesen Programmen. Unentbehrlich für jeden Grafikanwender. **ATARI ST Grafikanwendungen, ca. 200 Seiten, DM 29,–**

Ein Buch für jeden, der unter GEM Programme erstellen will! Arbeiten mit der Maus, Icons, Virtual Device Interface, Application Environment System und Graphics Device Operating System. Ein besonderer Schwerpunkt liegt im Einbinden von GEM-Routinen in C und 68000-Assembler und der Programmierung in diesen Sprachen. GEM – das Betriebssystem der Zukunft!
Das große GEM-Buch zum ATARI ST, 459 Seiten, DM 49,–

Was – Sie wissen nicht was DFÜ ist? Dann müssen Sie dieses Buch lesen! Es führt Sie umfassend in die Welt der Datenübertragung ein: Grundbegriffe, Soft- und Hardware für die eigene Mailbox, Akustikkoppler zum Selbstbauen, notwendige Schnittstellen und Kosten der DFÜ. Hacker sollten zum Schluß die Kapitel über rechtliche Bestimmungen, Datenschutz und Copyright lesen!
DFÜ für Jedermann zum COMMODORE 64 & 128, über 250 Seiten, DM 39,–

Lassen Sie sich verzaubern! Durch die Grafikmöglichkeiten des C-128. Aus dem Inhalt: die 3 Betriebsmodi, Grafikbefehle des BASIC 70, Textgrafik, Hi-Res/MC-Grafik, Sprites/Shapes, der VIC II und der VDC-Chip, Statistik, Funktionsplotter, CAD, Ein/Ausgabe von Grafiken, farbige hochauflösende VDC-Grafik, Grafikprogrammierung in 8502 Assembler u.v.m.
Das große Grafikbuch zum C-128, mehr als 350 Seiten, DM 39,–

Jetzt gibt es das große Floppybuch auch zur 1570/1571! Mit einer Einführung für Einsteiger, Arbeiten mit dem C-128 und BASIC 7.0, einer umfassenden Einführung in das Arbeiten mit sequentiellen und relativen Dateien, Programmierung für Fortgeschrittene: Nutzung der Direktzugriffsbefehle, Programme im DOS, wichtige DOS-Routinen, und natürlich ein ausführlich dokumentiertes DOS-Listing.
Das große Floppybuch zur 1570/1571, 583 Seiten, DM 49,–

Alles über Floppyprogrammierung vom Einsteiger bis zum Profi. Natürlich mit ausführlicher Kommentierung der DOS-Routinen, einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen!
Das Floppy-Buch zum CPC, 353 Seiten, DM 49,–

Tips & Tricks für alle CPC Benutzer! Menuegenerator, Maskengenerator, BASIC-Befehlserweiterungen, Programmierhilfen wie Dump, BASIC-Zeile von BASIC aus erzeugen, wichtige Systemroutinen und deren Nutzung, nützliche Routinen des BASIC-Interpreters, Beschleunigung von Programmen, relokative Maschinenprogrammierung u.v.m.
CPC Tips & Tricks Band II, 250 Seiten, DM 39,–

Um MS-DOS/PC-DOS wirklich nutzen zu können, muß man die umfangreichen Begriffe, Befehle und Strukturen kennen: Betriebssystem, Grundkommandos, der Editor EDLIN, Dateikommandos, Hierarchische Dateistrukturen, sequentielle, Random-, Index- und Kommandodateien und DEBUG-Kommandos. Dieses Buch verhilft Ihnen zum perfekten Umgang mit MS-DOS/PC-DOS! **Trainingsbuch zu MS-DOS/PC-DOS, 389 Seiten, DM 49,–**

DATA WELT 4/86

Randvoll mit Superartikeln zu ATARI ST, COMMODORE, CPC. Großer ST-Softwareführer, jede Menge Quicktips und aktuelle Tips & Tricks.

DATA WELT 4/86 ab 17. März am Kiosk

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name und Adresse bitte deutlich schreiben

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

Tagung für UNIX-Fachleute

Auf der Jahrestagung der GUUG (German UNIX Users Group), vom 7.-10. Oktober 1986, geben sich die UNIX-Fachleute ein Stelldichein. Im Rahmen einer Hard- und Software-Ausstellung mit Fachvorträgen und Industrieprogramm, das einzelne Anbieter zu Wort kommen läßt, wird reger Erfahrungs- und Informationsaustausch erwartet.

Network GmbH, Wilhelm-Suhr-Str. 14, 3055 Hagenburg, 0 50 33/70 57

CAT'86 wird größer

Die bereits angekündigte CAT'86, die vom 13.-16.Mai in Stuttgart stattfindet, hat wegen der großen Anzahl vorliegender Ausstellerbuchungen eine dritte Messehalle dazugemietet. Das Ausstellungsangebot dreht sich schwerpunktmäßig um Entwicklungen im CAD/CAM-Bereich.

Stuttgarter Messe- und Kongress GmbH, Postfach 990, 7000 Stuttgart 1, 0007 11/25 89-440

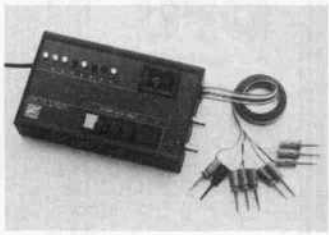
Trends und Prognosen

Über den Informations-Marktplatz der 'Micro-Computer'86', die am 2. Februar in Frankfurt zu Ende ging, zogen insgesamt 32 000 Besucher. Die klare Trennung in Fachbesucher- und Publikumstage wurde allgemein sehr positiv aufgenommen. Sinkende Preise für PCs, Leistungssteigerung der Homecomputer, wachsende Nachfrage nach Mehrplatzsystemen, Verbreitung der 20-MB-Festplatte im PC-Bereich, verbesserte 'Handcomputer' mit 3,5-Zoll-Diskettenlaufwerken, preiswertere Laserdrucker, neue Wechselplatten mit 10-20 MB Speicherkapazität und optische Platten (CD-Systeme) als Datenträger – das sind einige Trends, die der Messebesucher in Frankfurt erkennen konnte. Ganz deutlich wurde die Tendenz einer Schwerpunktverlagerung auf das Gebiet der Software: gefragt sind Spezialanwendungen, Branchenlösungen und Betriebsberatung. Eine Besucherbefragung ergab, daß der Anwender größeren Wert auf organisierte Software als auf technische Neuerungen legt. Zum nächsten Messetermin soll auch den 'Freaks' mehr Beachtung geschenkt werden, denn in Wirtschaftskreisen hat es sich herumgesprochen, daß 'immer mehr 'Turnschuhprogrammierer' in Bereiche vorstoßen, die kommerziell interessant sind'.

Messe Frankfurt GmbH, Ludwig-Erhard-Anlage 1, 6000 Frankfurt am Main, 0 69/75 75-477/520

Logik-Analyser

Ein Logik-Analysator für kleine Geldbeutel ist der Microlyser von Fotronic. Der Preis von unter 1000 DM ist dadurch möglich, daß Logikpegel und Taktraster von acht LEDs und einer zweistelligen Sieben-Segment-Anzeige dargestellt werden. Auf acht Kanälen können TTL-Signale mit einer von 10µs bis 100ms in fünf Schritten einstellbaren Auflösung und einer Speichertiefe von 100 Schritten analysiert werden. Die gespeicherten Daten lassen sich entweder über die LEDs abrufen oder mit einem angeschlossenen Oszilloskop auswerten. Die Vorgabe eines Triggerwortes ist ebenso möglich wie die Zuführung eines externen Taktes.



Fotronic GmbH, Josef-Beiser-Str. 13, 8000 München 83, 0 89/6 37 21 90.

Computer-Kunst auf der C'86

Auf der C'86-Computer, Software, Electronic, die vom 12.-15.Juni in Köln stattfindet, werden die besten Computer-Grafiken des Wettbewerbs 'Prints und Plots' ausgestellt. Das künstlerisch und technisch interessanteste Werk wird mit dem 'Goldenen Plotter' ausgezeichnet.

KölnMesse, Abt. MA II/2, Postfach 21 07 60, 5000 Köln 21

IBM-BRUTTOPREISLISTE für HÄNDLER 7/85

MICROMINT
– Zukünftiges heute –

B 1	FERTIGPLATINE	MICROMINT 16 XT-640 K-Mainboard mit 256 K bestückt, schnelles Bootrom (Copyright by MICROMINT) 1a Qualität, voll gesockelt, 8 vergoldete Slots, 8088 Prozessor, 8087 Optional, Taktfrequenz 4,77 MHz	429,-
B 2	GEHÄUSE	Klappgehäuse, verstärkte Metallausführung, Kunststoffblenden, kpl. mit Montagematerial	119,-
B 3	TASTATUR	Kapazitiv, programmierbar, 100% entprellt, Höhe verstellbar, ausschließlich deutsche Ausführung	235,-
B 4	NETZTEIL	150 Watt	215,-
B 5	NETZTEIL	130 Watt	195,-
B 6	LAUFWERK	MITSUBISHI: 2 x 40 Track, Head-Select, Automatischer Diskettenauswurf	319,-
B 7	HARDDISK	MITSUBISHI: 20 MB formatiert, incl. Kabel	1.490,-
B 8	MONITOR	ZENITH: TTL Eingang, 22 MHz, Bernstein entspiegelt	339,-
B 9	MONITOR	PHILLIPS: Video Chinch, 22 MHz, Grün entspiegelt	299,-
B 10	FARBMONITOR	ZENITH: 20 MHz, RGB-Anschluß, 13 Zoll	850,-
B 11	DISK-CONTROLLER	Für bis zu 4 Laufwerke	99,-
B 12	HARDDISK-CONTROLLER	Original Western Digital Controller 2 x 30 MB	490,-
B 13	KABEL-C	Anschlußkabel für Floppy-Laufwerke	25,-
B 14	KABEL-HD	Anschlußkabel für Harddisk-Controller	29,-
B 15	MULTIFUNKTION-CARD	1 x seriell, 1 x parallel, Game, Port, Uhr, mit 256 K bestückt	389,-
B 16	MULTI I/O-CARD	Ohne Ram-Bestückung, 1 x seriell, 1 x parallel, Game-Port, Uhr, Disk-Controller-Anschluß	299,-
B 17	COLOR-GRAPHIK-CARD	RGB- und BAS-Signal, Chinch-Anschluß	159,-
B 18	HERCULES-CARD	Hochauflösende Monochrom-Karte 720 x 340 Bildpunkte TTL-Ausgang	279,-
B 19	JOY-STICK	passend für IBM, ATARI, COMMODORE	29,-
B 20	MONITORDREHFÜSSE	passend für alle Monitore, hochwertige Kunststoffausf.	17,-

Aufwärts durch Qualität

Bei höheren Abnahmemengen setzen Sie sich bitte mit unserem Verkaufsteam in Verbindung – wir beraten Sie gerne!
Wir liefern Ihnen jede gewünschte Menge in kürzester Zeit ab Lager.

MICROMINT

1000-fach bewährt!
MICROMINT 16
100% XT-IBM® comp.

- modernste System-Architektur
- Micromint high-speed Bootrom: Bootzeit 20 sec. Testbereich: 3-fach erweitert
- 100% IBM®-XT Comp.
- Bis 640 K aufrüstbar mit 256 K bestückt
- 8088 Prozessor 8087 Optional
- Taktfrequenz 4,77 MHZ
- vergoldete Slots
- Tastatur: kapazitiv, programmierbar ausschließlich deutsch
- Monitor: wahlweise mit TTL, Video-chinch oder RGB-Ausgang
- Laufwerke: Mitsubishi 2 x 40 Track-Slim-Line Mitsubishi Harddisc 20 MB formatiert
- Schnittstellen: 1 x parallel, 1 x seriell, Gameport, Uhr
- Disc-Controller, Harddisc-Controller, Hercules-Card, Multi I/O-Card
- Netzteil: Wahlweise 130 Watt oder 150 Watt gegen Aufpreis.

MICROMINT Computer GMBH
Hochdahler Str. 151 · D-4006 Erkrath 2 (Hochdahl)
Postgirokonto Essen 2761 51-430 (BLZ 360 100 43)

02104/33024
Telex-Nr.: 8 589 305 mcmd

2 NEUE TOOLBOXEN FÜR TURBO 3.0 FANS

TURBO GAMEWORKS

Schach, Bridge und GoMoku – die drei Klassiker der Karten- und Brettspiele als (zugegebenermaßen nicht ganz vollwertiger) Ersatz für klassische Bettspiele.

Dabei ist Spielen mit diesem Paket noch das kleinste Vergnügen: Indem Sie Ihr eigenes Spiel verbessern und dem vorhandenen Schach (Bridge/GoMoku) Variationen hinzufügen, lernen Sie, wie man in Turbo-Pascal Spiele programmiert. Spaß und Unterhaltung halten sich die Waage mit dem Lerneffekt, den Sie mit diesem Programm wie nebenbei erzielen. Es ist deshalb auch vorzüglich zur Demonstration und für den Unterricht geeignet. Turbo-Gameworks ist unsere vergnüglichste Toolbox aus der Turbo-Pascal-Family und sollte deshalb jedem Anfänger mitgegeben werden. Der Profi hat Gelegenheit, sich in die Höhen der Spieltheorie zu versteigen und wer gar nicht tun will, läßt seinen Computer gegen sich selbst spielen. Das kann Ihr PC zur Erholung auch mal brauchen. Vorläufig brauchen Sie aber einen IBM PC (oder Kompatiblen) zum Spielespaß und natürlich zum Compilieren Turbo-Pascal 3.0

Für Kenner ein paar Daten, was mit Gameworks möglich ist:

- *jederzeit ein Spiel abbrechen und später wiederaufnehmen.*
- *die Spielstärke stufenlos variieren*
- *beliebige Stellungen eingeben und analysieren*
- *Zugvarianten durchspielen*
- *spezielle Modi für Mattaufgaben und Austesten neuer Strategien*
- *Auswertung eines ausführlichen Protokolls.*

Zitat: *Georg Koltanowski, Schach-Kolumnist und Präsident der amerikanischen Schachvereinigung: »Spaß und Unterhaltung bei TURBO GAMEWORKS sind grenzenlos und vor allem kann das Programm dazu beitragen, Ihr Schachspiel unbegrenzt zu verbessern«.*

TURBO EDITOR

In dieser Toolbox finden Sie wirklich alles, was Sie sich zur Umsetzung Ihrer Texte auf einem PC nur wünschen können:

MicroStar – ein unglaublich leistungsfähiges Textverarbeitungsprogramm mit allem Drum und Dran:

- *Pull-Down Menü, die komfortabelste und leichteste Art, ein Programm zu bedienen.*
- *Windows, bis zu 8 verschiedene Texte, können Sie durch 8 Bildschirmfenster gleichzeitig ansehen, verändern und montieren.*
- *Multitasking, während Sie den fertigen Text ausdrucken wollen, schreiben Sie bereits den nächsten – gleichzeitig!*
- *RAM-resident, auch in großen Dateien blättern und arbeiten Sie ohne jedes Warten.*
- *Lightning-Schnittstelle, schon heute können Sie MicroStar oder Ihr Turbo-Programm an Borlands neuester »Sensation« (Zitat aus PC Magazine) TURBO-LIGHTNING anschließen! Turbo-Lightning ist eine blitzschnelle Rechtschreibkorrektur, aber auch Sprachtraining durch Anzeige bedeutungsähnlicher Wörter. Vorläufig in englisch.*
- *und natürlich mit allen Funktionen der großen Textverarbeitungssysteme wie UNDO, Block, Zentrieren, Tabulieren, Umbruch etc. ausgestattet.*

MicroStar™ wird als funktionsfähiges Programm und im Turbo-Pascal Quellcode geliefert. MicroStar™ ist also nicht nur ein sehr edles Textsystem, sondern auch eine Riesenfundkiste für die Konstruktion und die Benutzeroberfläche Ihrer eigenen Turbo-Programme. Freie Nutzung ohne Lizenzgebühren! Vorläufig nur für IBM-PC und Kompatible, zum Compilieren brauchen Sie natürlich Turbo-Pascal 3.0!

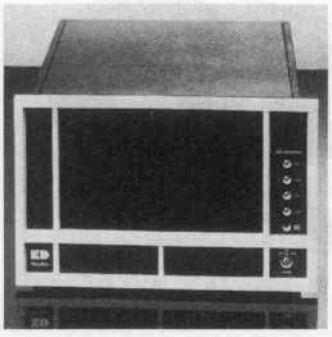
Jede Toolbox wie immer DM 225,72 **inclusive 14% Mehrwertsteuer.**

HEIMSOETH software
Fraunhoferstr. 13 · D-8000 München 5
Telefon D-089 / 26 40 60 / 260 85 81
Telex mcm 5 212 637

Leiterplatten-CAD mit PCs

Auf dem PC/AT mit mindestens 256 KByte Arbeitsspeicher und zwei Floppy-Laufwerken läuft EDMAS, eine aus Grafikrechnerkarte, Joystick und CAD-Software bestehende Kombination zur Entwicklung von Leiterplatten mit bis zu 20lagigen Multilayern. Das System unterstützt nicht nur das Erstellen von Schaltplänen und den Entwurf des Leiterplattenlayouts, sondern auch das Anfertigen von Konstruktionszeichnungen für mechanische Teile. Die 100prozentige Übereinstimmung von Schaltplan und Layout ist durch die Verwendung eines für alle Prozesse gemeinsamen Datenbestandes gewährleistet. Zusätzlich kann EDMAS Daten für Foto-Plotter und NC-Maschinen bereitstellen sowie alle üblichen Fertigungsunterlagen wie Bestückungsplan, Lötstopmaske, Bohrplan, Bemaßungsplan und Stückliste. Der Preis für EDMAS liegt bei 30 000 DM.

Questos Computer GmbH, Am Bogen 1, 8012 Ottobrunn, 089/609 70 89



Träger für Multibus-Karten

Einen Multibus-Kartenträger im Metallgehäuse bietet Key Design an. Die Micro-Box 'MB1-12 mit 12 Multibus-Steckplätzen, Einschubrahmen, mehreren Lüftern und Frontplatte in Einschnapp-Technik hat unter dem Kartenträger noch Platz für zwei halbhohe 5,25-Zoll-Laufwerke, weitere Massenspeicher oder Netzteile. Der als Tischgehäuse oder 19-Zoll-Einschub verwendbare Kartenträger inclusive Netzteil mit 235 Watt ist für knapp 7000 DM lieferbar.

Rauscher GmbH, Bodenseestr. 293, 8000 München 60, 0 89/87 80 61

Erweiterte CAD-Software

Dedata hat die CAD-Software COMP-U-DRAFT um das Modul VARICAD erweitert und 'DedataCAD' getauft. VARICAD ist ein CAD-System, mit dem Anwender ohne Programmierkenntnisse arbeiten können. Insbesondere wird die Unterstützung vor- oder nachgeschalteter benutzerspezifischer Programme ermöglicht. Das 'offene' CAD-System ist während der CeBIT-Messe in Halle 17/Stand 101 D zu besichtigen.

dedata edv, POB 2008, 7500 Karlsruhe 1, 07 21/84 64 2

Datenaustausch mit Atari ST

Wichtige Adreßdateien, Manuskripte, Briefe, Programm-Listings und so weiter brauchen nicht mehr neu eingetippt zu werden. Das GEM-Kommunikationsprogramm 'S-Term plus' für den Atari ST macht den Datenaustausch mit anderen Computern per RS-232-Verbindung möglich und unterstützt außerdem Mailbox- und andere Modem-Kommunikationsarten. Der Preis für das Programm mit ausführlichem deutschen Handbuch beträgt 199 DM. Für zusätzliche 20 DM wird eine Demo-Diskette mitgeliefert.

Harald Zoschke, Postfach 1264, 8150 Holzkirchen, 0 80 24/3592

IEEE488-Interface

Das Interface 38000/B erlaubt, Peripheriegeräte mit IEEE488-Schnittstelle an IBM PCs und Kompatible anzuschließen. Es ist als intelligentes, externes Gerät konzipiert und arbeitet unabhängig vom Betriebssystem des Rechners. Der Anschluß erfolgt über die V.24-Schnittstelle. Alle Adressen und IEEE-Kommandos können zum Beispiel von BASIC aus gegeben werden, um etwa IEEE-Meßgeräte vom PC aus anzusprechen. Das Gerät ist für 1117 DM erhältlich.

Wiesemann Mikrocomputertechnik, Winchenbachstr. 3-5, 5600 Wuppertal 2, 02 02/50 50 77.

ATARI ST !

ATARI 260 ST / 520 ST+ lieferbar - Preise auf Anfrage!
3,5" Floppy CH 32 mit 720 KB formatiert, Netzteil im Gehäuse 698,--
3,5" Doppel-Floppy CH 62 mit 2 x 720 KB, Netzteil im Gehäuse 1.198,--
3,25" Floppy CH 52 mit 720 KB formatiert, Netzteil im Gehäuse 798,--
3,25" Doppel-Floppy CH 102 mit 2 x 720 KB, Netzteil im Gehäuse 1.298,--
Speichererweiterung auf 1 MByte RAM für ATARI 260/520 350,--
Speichererweiterung auf 2 MByte CMOS-RAM für alle ATARI ST ab April
mit 1 MByte, batteriegepuffert, komplett mit RAM-Disk Software
EPROM-Programmiergerät 387,80
für EPROM-Typen 2764, 27128, 27256 + entsprechende C-MOS Typen
VIDEO-DIGITIZER 698,--
Video-Digitizer zum Abspeichern, Ausdrucken und Weiterbearbeiten von Video-Bildern. Demo-Ausdruck liegt bei.
Professionelles PC-Gehäuse für den ATARI ST 198,--
Staubschutzhaube für alle ATARI ST 30,--
ST-Monitorständer 69,--
Programmiersprachen:
GST ASM Macro Assembler 248,--
GST C C Compiler 348,--
GST EDIT Screen Editor 148,--
FOCUS UCSD p-System mit UCSD Pascal Compiler 890,--
FOCUS FORTRAN-77 Compiler unter UCSD p-System 990,--
FOCUS BASIC Compiler unter UCSD p-System 790,--
p-System Tool Kit 490,--
mit Debugger, Native Code Generator, Assembler, etc.
Metacomco Pascal Compiler 475,--
Metacomco Assembler 395,--
The Dragon Group 4 x FORTH
Level 1, bei uns bereits mit Nachtrag! 498,--
Level 2 750,--
Upgrade Level 1 - Level 2 300,--
Accelerators für 4 x FORTH Level 1 und 2 398,--
Kuma K-SEKA - schneller 68000-Assembler deutsch! 189,--
Software:
DEGAS - der Hammer! Graphik-(Zeichen-)Programm, sogar umfangreicher als GEM-Draw! Mit absoluter Maus/GEM-Unterstützung und englischer Bedienungsanleitung. 159,--
15 Demo-Disketten randvoll mit Original ATARI Programmen, darunter "NEOchrome", ein ATARI Farb-Graphik-Programm! 149,--
CP/M-280-Emulator 29,--
C-adress/C-text Adressenverwaltung und Textverarbeitung 285,--
C-auftrag/C-text Auftragsabwicklung und Textverarbeitung 570,--
MICROWAVE MAXITEXT, Text-Editor mit deutschen Bedienerhandbuch 398,--
NEU! S-TERM + Kommunikations-Programmpaket 198,--
VIP PROFESSIONAL, LOTUS-kompatibles Integriertes Tabellenkalkulationsprogramm, englisch, mit deutschem Handbuch 748,--
VT 100-Emulator 349,-
JJC-DISKMON - Ein professioneller Disketten-Monitor, der auf sehr komfortable Weise einen direkten Zugriff auf alle Daten der Diskette erlaubt.
JJC-CALCULATOR 59,--
Desk Accessory SPOOLER Druckerspooler 149,--
Desk Accessory PROGRAMMER'S CALCULATOR, mit den zusätzlichen logischen Operationen AND, OR, XOR, usw. Beherrscht zusätzlich auch hexadezimale, oktale und binäre Zahlen. 179,--
Kuma K-RAM - RAM-Disk deutsch! 119,--
Kuma K-SPREAD - ein professionelles Tabellenkalkulationsprogra 229,--
Hitchhiker's Guide to Galaxy Adventure Spiel 189,--
Lands of Havoc 89,--
Skull/West Adventure Spiel 129,--
Oasis Systems Sundog (Spiel) 149,--
Mark of the Unicorn Software:
Mince Text Editor 499,--
PC-Intercom VT-100 Emulator 399,--
Hex (Q-Bert) 149,--

Genie 286 AT, CPU 80286, 512k RAM, MS-DOS 3.1 deutsche Tastatur, 20MB-HD, 1,2MB Floppy, Uhr
Genie 16 mit 640k RAM, neuer, 30% schnellerer CPU, MS-DOS 2.11, + deutsche Textverarbeitung
Genie 16 C mit interner HD 10MB, 1 Laufwerk 256k RAM, 7,2 MHz, 2 Laufwerke mit je 720k incl. Monitor
Backup-Streamer 10MB Streamline, Ansteuerung mit Floppy-Contr. Gesonderte PHOENIX Preisliste anfordern!
Genie 286 AT
Genie 16 C
Genie 16 XC
Genie III s

DM 9.795,--
DM 4.495,--
DM 7.495,--
DM 6.500,--
DM 1.995,--

Computershop Habersaat

COMPUTERSHOP HABERSAAT
Inh. Wolfgang Habersaat
BAHNHOFSTR. 12
8087 TURKENFELD
TEL. 08193/6031

W+H COMPUTER-HANDEL Gmbh
FÖRSTERGASSE 6/3/2
A-1020 WIEN
TEL. 0222/3384592

Ultima III Adventure Spiel auf Anfrage
Kings Quest Adventure Spiel auf Anfrage
Hippo "C" Programmiersprache "C" auf Anfrage
Mitchtron Mudpies 89,--
Mitchtron Flipside (Reversi) 89,--
Mitchtron RAM-Disk 149,--
Mitchtron Printer-Spooler 149,--
Crimson Crown Adventure Spiel 149,--
Transylvania Adventure Spiel 149,--
Hippopotamus ST Disk Utilities 199,--
Hippopotamus ST RAMDISK 149,--
Bücher:
CHIP-Sonderheft PC-Soft Anwender-Programme ATARI ST (Einstieg) 28,--
ATARI ST für Einsteiger 29,--
Von BASIC zu C mit dem ATARI ST 39,--
Das Prozessorbuch zum 68000 59,--
ATARI ST Tips & Tricks 49,--
ATARI ST Intern 69,--
ATARI ST Grafik und Sound 49,--
Das Maschinensprachebuch zum ATARI ST 39,--
Das große GEM-Buch zum ATARI ST 49,--
Das LOGO-Trainingsbuch zum ATARI ST 49,--
LOGO-Handbuch deutsch incl. Diskette mit Beispielen 29,--
Der ATARI 520 ST 49,--
GEM für den ATARI 520 ST 52,--
MIDI-KOMPENDIUM 2. Auflage 29,80
Basic-Programmierung mit dem ATARI 520 ST 29,80
Zubehör:
3,5" Disketten 1DD 5er-Pack weiße Ware 39,--
3,5" Disketten 1DD 10er-Pack Fuji 99,--
5,25" Disketten 1D 10er-Pack weiße Ware 29,--
multiform Ablagekasten mit Deckel für 40 Disketten 3,5" 39,--

Fordern Sie sofort unser Info an!!!

Wir führen auch Zubehör für den TEXAS INSTRUMENTS TI 99/4A!
Z.B. RGB-Farbmonitor 14 MHZ mit Scart-Anschluß DM 598,--.

TAXAN – für die Welt von morgen.

TAXAN KP-810/910: professioneller Matrixdrucker. Schnell: 140 Zeichen/Sek. · leise: < 60 dB · Schönschrift: Near Letter Quality (NLQ) · variabel: frei programmierbarer Zeichensatz · ausbaufähig: Zusatzkarten mit verschiedenen Schriftarten/IBM PC-Zeichensatz. KP 810 – 80 Zeichen pro Zeile. KP 910 – 156 Zeichen pro Zeile. TAXAN – denn die Zukunft hat bereits begonnen.

Ausführliche Information erhalten Sie beim Fachhandel.

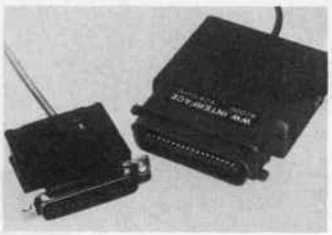
Mikrocontroller mit programmierbarem Datenbus

Programmierbar ist die Datenbusbreite (8 Bit oder 16 Bit) des Mikrocontrollers 8096BH aus der MCS-96-Familie von Intel. Damit kann er auch in 8-Bit-Designs eingesetzt werden. Er vereinigt in einem Chip eine 16-Bit-CPU, schnelle Multiplizier- und Dividier-Subsysteme, serielle Ports und einen 10-Bit-A/D-Wandler und eignet sich für I/O-intensive Echtzeitanwendungen wie beispielsweise Motor- und Robotersteuerungen. Die 8-KByte-EPROM-Ausführungen mit der Bezeichnung 8796BH kostet 75 Dollar (1000-Stück-Preis). Als vielseitiges Testhilfsmittel für Hard- und Software ist der In-Circuit-Emulator VLSiCE-96P für Mikrocontroller der MCS-96-Familie erhältlich. Er arbeitet auch mit dem PC/AT und dem PC/XT zusammen und kostet 11 500 Dollar. Ein Low-Cost-Entwicklungssystem für die MCS-96-Familie ist der Single-Board-Emulator iSBE-96. Er arbeitet unter anderem mit PC-DOS-kompatiblen Rechnern zusammen. Der Preis von 2500 Dollar schließt den Makro-Assembler ASM-96 mit ein.

Intel Semiconductor GmbH, 8000 München 2, Postfach 20 13 06, Seidelstr. 27, 089/53 89-1.

Drucker-Interface für Siemens PC-D

Mit dem Interface Typ 82000/PCD lassen sich an den Siemens PC-D alle gängigen Drucker mit Centronics-Schnittstelle anschließen. Dieses Interface von Wiesemann ist dem speziellen Handshake der Rechner angepaßt und wird zum Preis von 298 DM steckerfertig geliefert. Die Stromversorgung erfolgt entweder vom Drucker (Pin 18 der Schnittstelle) oder optional über ein separates Steckerteil. Für 348 DM ist das Gerät auch in der Version 82008/PCD mit 8-KB-Pufferspeicher erhältlich.



Reinhard Wiesemann, Winchenbachstr. 3-5, 5600 Wuppertal 2, 02 02/50 50 77

Z80 in CMOS

Den Mikroprozessor Z80 als kompatible CMOS-Version bietet nun auch SGS an. Zunächst sind diese Bausteine für Taktfrequenzen von 2,5 MHz und 4 MHz erhältlich – eine 6-MHz-Version wird angekündigt. Alle CMOS-Typen sind anschlußgleich und softwarekompatibel zu den bekannten NMOS-Versionen des Z80 und werden durch den neuen CMOS-Taktgenerator/Teiler M 8710 unterstützt. Sie sind im DIL-Kunststoff- und Keramikgehäuse sowie als Chip Carrier für die Oberflächenmontage erhältlich.

SGS Halbleiter Bauelemente GmbH, Haidling 17, Postfach 1180, 8018 Grafing bei München

Klare Sicht

Ein von der Kontakt-Chemie entwickeltes Bildschirm-Reinigungsspray hat den Vorteil, daß es durch Schaumbildung nicht auf dem Bildschirm verläuft und durch einen Antistatik-Effekt elektrostatische Aufladungen beseitigt und neue Staubanziehung verhindert. Das Spray ist in einer 150-ml-Dose für 9,90 DM erhältlich und eignet sich auch zur Reinigung von Brillengläsern.

Kontakt Chemie GmbH, Am Krebsbach 26, 7550 Rastatt-Niederbühl, 07 61/3 60 28.

PLOTTERKUNDE (nicht nur) FÜR PLOTTERKUNDIGE

Plotter-Kauf sollte nicht zum „PLOTTER-LOTTO" werden.

GRAPHTEC 24 MONATE GARANTIE

Wo aber finden Sie Plotterkundige?
Zum Beispiel bei GRAPHTEC.

GRAPHTEC bietet vom DIN A4- bis DIN A0-Plotter, in Flachbett-, Rollen- oder Trommelbauweise, ein komplettes Programm.

Fragen Sie uns – wenn es um Plotter geht.

REGISTRIERTECHNIK 35 JAHRE

GRAPHTEC

Wir stellen aus: Hannover CEBIT Halle 3, Stand 1407 · Hannover Industrie Halle 12, 1. OG., Stand 1401

Postfach 1155 · D-8036 Herrsching · Telefon (08152) 3084 · Teletex 815281 · Telefax (08152) 2521
Berlin (030) 8835063 · Hamburg (040) 459517 · Düsseldorf (0211) 721145 · Karlsruhe (07243) 90666 · Wien (0222) 973397

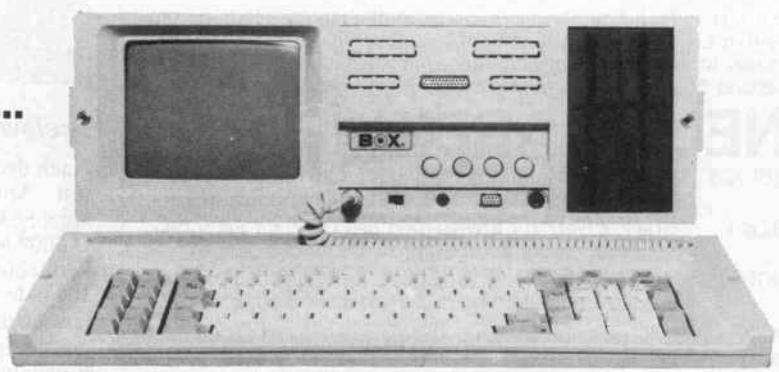
ECD – PROFESSIONAL IBM-XT-kompatibel

- **640 KB,** voll bestückt
- 1 Laufwerk
- Color-Grafik-Karte
- **150-W**-Netzteil
- DIN- oder ASCII-Tastatur
- Komplettpreis **DM 1962.–**

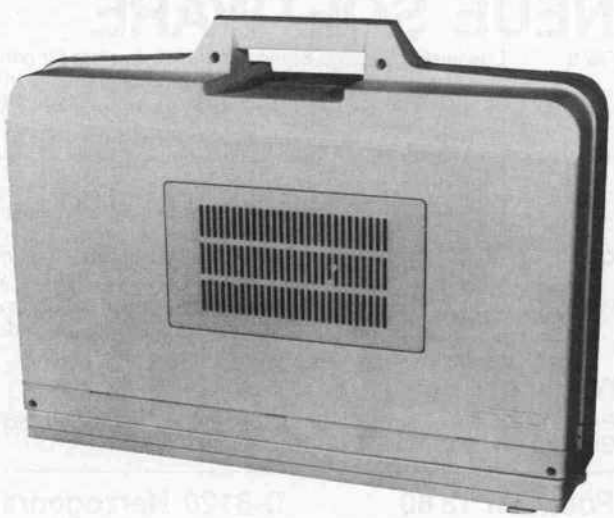
Preise inkl. MwSt. zuzügl. Versandkosten
IBM ist eingetragenes Warenzeichen

Erweiterungen:		
	Multi-I/O-Karte (Uhr/RS232/Parallel)	DM 345.–
	Monochrome-Grafik-Karte (Herkules-komp.)	DM 375.–
	Eprom-Burner	DM 495.–
	AD/DA-Wandler-Karte	DM 475.–
	TTL-Monitor (für Herkules komp.)	DM 398.–
	RGB-Farbmonitor 640 × 400	DM 1250.–
	Diskettenlaufwerk 360 KB	DM 365.–
	22-MB-Harddisk inkl. Controller	DM 2350.–

Unser Portable ist tragbar ...



- Nur 10 kg
- XT-kompatibel
- 640-KB-Mainbord (256k best.)
- 8 Erweiterungs-Slots
- 7 freie Eprom-Steckplätze
- 7″-Monitor (bernstein)
- 2 Stück 360-KB-Laufwerke
- Printeranschluß
- Inkl. MS-DOS 2.11
  Handbücher
  Service-Software
- Komplettpreis **DM 3198.–**



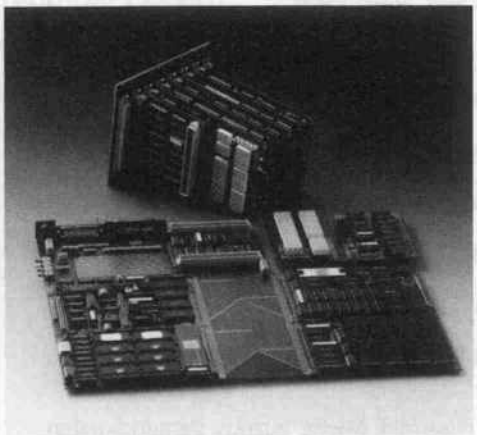
IHR PARTNER IN SÜDDEUTSCHLAND

ECD COMPUTERTECHNIK GmbH
HAINDLFINGER STRASSE 20
8051 ZOLLING/PALZING

BÜRO: 8050 FREISING, TEL. 0 81 61/1 37 39

GWK

GESELLSCHAFT FÜR TECHNISCHE ELEKTRONIK mbH
HARDWARE SOFTWARE SYSTEMENTWICKLUNG



c't 68000 GWK

Hochleistungscomputer mit CPU 68000 wie im c't magazin für Computertechnik ab Heft 10-84 beschrieben. Modulare Ausführung auf Europakarten.

NEUE PREISE

	Preis zuzügl. ges. MwSt.	**Preis incls. ges. MwSt.**
Komplettsysteme ab	5985,— DM	**6822,90 DM**
Fertigplatinensatz ab	2985,— DM	**3402,90 DM**
Bausätze ab	1885,— DM	**2148,90 DM**
Leerplatinensatz ab	590,— DM	**672,60 DM**

Die hier aufgeführten Konfigurationen enthalten jeweils den kompletten Leistungsumfang mit CPU, 256 KByte DRAM, Floppy Controller, Input/Output, Uhr und Timer mit Ausnahme der Grafik, die separat bestellt werden muß.

NEUE HARDWARE

SBI ICS — System Bus Interface für gepufferten 16 Bit Bus. Volle Interruptbehandlung. Compatibel mit c't Bus.

GZS 1 — Grafik Zusatz zur Erweiterung des GDP auf Farbgrafik. Acht Farben ohne Auflösungsverlust.

HDI — Hard Disk Interface. Ermöglicht den Anschluß von Winchester Controller und drei Winchester Laufwerken.

Weitere neue Karten sind in Arbeit. Ab sofort sind die c't 68000 Platinen auch einzeln erhältlich.

NEUE SOFTWARE

OS-9 — Das leistungsfähige Standard Betriebssystem für den c't 68000 ist jetzt verfügbar. Kernel, Shell Kommando Interpreter mit ca. 60 Utilities, Editor, Assembler, Linker und Debugger.

Sprachen C und PASCAL als Compiler und BASIC als Interpreter verfügbar.

Standardsoftware für Textverarbeitung, Tabellenkalkulation und relationale Datenbank lieferbar.

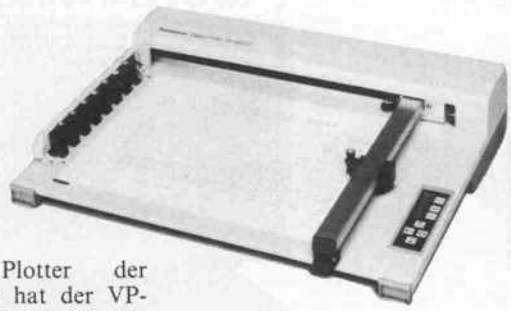
CP-M/68K — Betriebssystem CP-M/68K liegt jetzt in einer eigenen, auch in Zukunft voll unterstützten Implementierung vor.

RTOS — Verbesserte neue Version mit einer für den Anwender zugänglich gemachten Input/Output Ebene. Einbindung eigener Treiber wird ermöglicht.

Fordern Sie ausführliches Informationsmaterial an!

Preise verstehen sich zuzüglich Verpackung und Versandspesen. Garantie auf Fertigplatinen und Systeme 6 Monate.

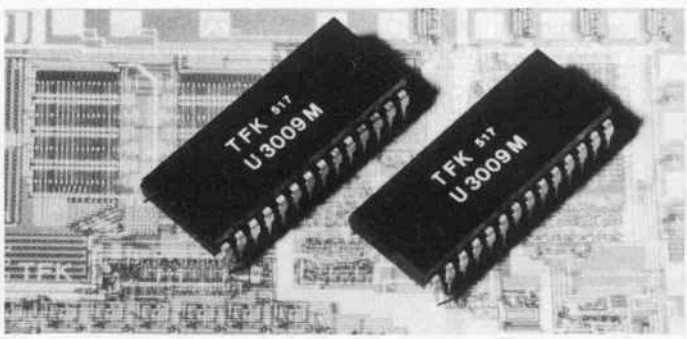
Postfach 13 60 — D-5120 Herzogenrath
Telefon 0 24 06/60 35 — Telex 8 32 109 gwkd



A3-Plotter preiswert

Für einen Plotter der 3500-DM-Klasse hat der VP-6803 P von National Panasonic immerhin acht Farben und eine Geschwindigkeit von 200 mm/s zu bieten. Die Schrittweite ist programmierbar auf 0,05 mm und 0,025 mm, wobei alle Koordinatenwerte automatisch auf 0,05 mm gerundet werden. Den Gesamtfehler einschließlich Stiftwechsel gibt der Hersteller mit weniger als 0,2 mm an. Die Ansteuerung des Plotters geschieht seriell mit bis zu 9600 Baud. Die Schnittstelle läßt sich mittels DIL-Schalter an verschiedene Übertragungs-Formate anpassen. Die Software des Plotters ist HP-GL-kompatibel, was ihn beispielsweise befähigt, mit den Programmpaketen Lotus und Symphonie zusammenzuarbeiten.

Dynatrade Electronic GmbH, Schimmelbuschstr. 25, 4006 Erkrath 2, 0 21 04/3 11 47



Preiswerter A/D-Wandler

Nach dem Prinzip der sukzessiven Aproximation geht der Analog-Digital-Umsetzer U 3009 M von Telefunken vor. Bei einer Taktfrequenz von 100 kHz schafft er eine 8-Bit-Umsetzung in 160 µs. Mit integriert sind bereits ein 4-Kanal-Eingangsmultiplexer, eine Sample & Hold-Stufe und ein Taktoszillator. Die Daten können parallel oder seriell ausgegeben werden. Der Wandler im 28poligen DIL-Gehäuse kann analoge Eingangsspannungen bis zur Höhe der Betriebsspannung von 5 V verarbeiten. Erhältlich ist der Baustein mit einer absoluten Genauigkeit von ±0,4%, ±0,8% und ±1,6% für 18 DM.

Telefunken electronic GmbH, 7100 Heilbronn, Postfach 1109, 0 71 31/670

Notstrom für Computer

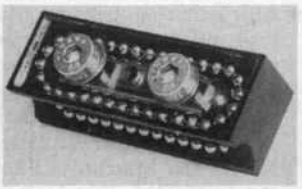
Das USV-Gerät 'Microstal' gewährleistet unterbrechungsfreie Stromversorgung für Mikro-Computer. In der eingebauten Batterie wird der Netzstrom gleichgerichtet und über einen Wechseltrichter an den Computer geliefert. Je nach Computersystem lassen sich Ausfälle von bis zu 15 Minuten überbrücken. Imunelec bietet das Gerät in zwei Ausführungen an: MS 250 (130*388*287mm) und MS 600 (390*195*465mm). Die Preise betragen ca. 3000 und 4200 DM.

Imunelec GmbH, Achenbachstr. 2, 6000 Frankfurt/Main 70, 0 69/63 60 01

isert-electronic

„isel"-Präzisions-Linearkugellager DM 15,70

- Präzisions-Linearkugellager, Außenm. L 60 x B 20 x H 17 mm
- 2 Deckplatten, L 57 x B 18 x H 3 mm, gehärtet u. geschliffen
- Zwei parallele Kugellaufbahnen, Kugeldurchmesser 3,5 mm
- Bolzen Ø 12 mm, H6 L 50 mm, gehärtet und geschliffen
- Spielfreie Einstellung und Befestigung mit M6er-Schrauben
- Glasfaserverstärkter Kunststoffkäfig für Kugellaufbahnen



„isel"-Linear-Doppelspurset (o. Abb.) DM 37,50

- 2 Präzisions-Linearkugellager an Alu-T-Nutenprofil
- Spielfrei einstellbar und befestigt mit M6er-Schrauben
- Alu-Befestigungsplatte m. Alu-Aufspannfläche, L 60 x B 72 mm
- Verdrehsichere Präzisionsführung für 12er-Stahlwellen

Schrittmotor Nr. 3450 (ohne Abb.) **DM 113,00**
- Zweiph.-Schrittmotor, Schrittw. 1,8 Grad, Haltem. 55 Ncm

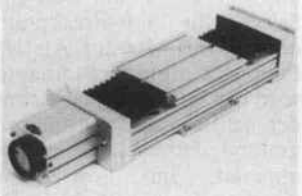
Schrittmotor Nr. 3455 (ohne Abb.) **DM 168,00**
- Zweiph.-Schrittmotor, Schrittw. 1,8 Grad, Haltem. 110 Ncm

Getriebeschrittm. Nr. 3463 (ohne Abb.) **DM 225,00**
- Zweiph.-Schrittmotor, Schrittwinkel 1,8 Grad, Untersetz. 1:9

Getriebeschrittm. Nr. 3465 (ohne Abb.) **DM 225,00**
- Zweiph.-Schrittmotor, Schrittwinkel 1,8 Grad, Unters. 1:50

„isel"-Vorschubeinheit 1 mit Schrittmotor und Kugelgewindetrieb DM 698,00

- Präz.-Vorschubeinh. m. T-Nuten-Profil, L 475 mm, B 125 mm
- Verfahrweg max. 250 mm mit 4 „isel"-Linear-Kugellagern
- Zwei Linear-Wellenklemmprofile mit 12er-Stahlwellen
- 1 Kugelgewindetrieb, Ø 12 x 2 mm, mit 2 Flanschlagern
- 1 Schrittmot., Schrittwinkel 1,8 Grad, Haltemoment 55 Ncm
- T-Nuten-Aufspannplatte 150 x 125 x 15 mm, spielfreier Hub
- 1 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

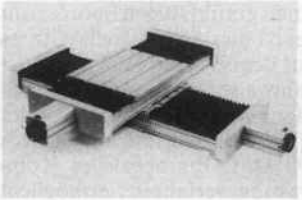


„isel"-Vorschubeinheit 2 (o. Abb.) mit Schrittmotor u. Kugelgewindetrieb DM 980,00

- Präz.-Vorschubeinh. m. T-Nuten-Prof., L 775 mm, B 250 mm
- Verfahrweg max. 400 mm, mit 4 „isel"-Linear-Kugellagern
- Zwei Linear-Wellenklemmprofile mit 12er-Stahlwellen
- 1 Kugelgewindetrieb, Ø 12 x 2 mm, mit 2 Flanschlagern
- 1 Schrittmot., Schrittwinkel 1,8 Grad, Haltemoment 110 Ncm
- T-Nuten-Aufspannplatte 285 x 250 x 15 mm, spielfreier Hub
- 1 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

„isel"-x/y-Kreuztisch 1 (o. Abb.) DM 1380,00 2 Schrittmotore u. 2 Kugelgewindetriebe

- Präzisions-x/y-Kreuztisch mit 2 Linear-Vorschubeinheiten
- Verfahrweg in x-Richt. 150 mm und in y-Richt. 250 mm
- Vorschübe mit „isel"-Linearlagern auf 12er-Stahlwellen
- 2 Kugelgewindetriebe, Ø 12 x 2 mm, mit 4 Flanschlagern
- 2 Schrittmot., Schrittwinkel 1,8 Grad, Haltemoment 55 Ncm
- T-Nuten-Aufspannplatte 150 x 125 x 15 mm, spielfreier Hub
- 2 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

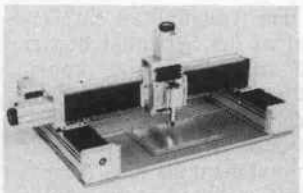


„isel"-x/y-Kreuztisch 2 DM 1930,00 2 Schrittmotore u. 2 Kugelgewindetriebe

- Präzisions-x/y-Kreuztisch mit 2 Linear-Vorschubeinheiten
- Verfahrweg in x-Richtung 250 mm und in y-Richtung 400 mm
- Vorschübe mit „isel"-Linearlagern auf 12er-Stahlwellen
- 2 Kugelgewindetriebe, Ø 12 x 2 mm, mit 4 Flanschlagern
- 2 Schrittmot., Schrittwinkel 1,8 Grad, Haltemoment 110 Ncm
- T-Nuten-Aufspannplatte 285 x 250 x 15 mm, spielfreier Hub
- 2 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

„isel"-x/y/z-Anlage 3 DM 2680,00 3 Schrittmotoren u. 3 Kugelgewindetriebe

- Präzisions-x/y-Koordinatentisch mit „isel"-Linearführung
- Verfahrweg in x-Richt. 300 mm u. in y-Richt. 400 mm
- Alu-T-Nuten-Profil mit Aufspannfläche 500 x 550 mm
- Präzis.-z-Achse, Hub 80 mm, mit Linear-Vorschubeinheit
- 2 Schrittmotor, Schrittwinkel 1,8 Grad, Haltemoment 110 Ncm
- 1 Schrittmotor, Schrittwinkel 1,8 Grad, Haltemoment 55 Ncm
- 2 Kugelgewtr., Ø 16 x 4 mm, 1 Kugelgewtr., Ø 12 x 2 mm
- 3 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm



„isel"-x/y/z-Anlage 4 (ohne Abb.) DM 3700,00 3 Schrittmotoren u. 3 Kugelgewindetriebe

- Präzisions-x/y-Koordinatentisch mit „isel"-Linearführung
- Verfahrweg in x-Richt. 600 mm und in y-Richt. 600 mm
- Alu-T-Nuten-Profile mit Aufspannfläche 1000 x 800 mm
- Präzis.-z-Achse, Hub 80 mm, mit Linear-Vorschubeinheit
- 2 Schrittmotor, Schrittwink. 1,8 Grad, Haltemoment 110 Ncm
- 1 Schrittmotor, Schrittwinkel 1,8 Grad, Haltemoment 55 Ncm
- 2 Kugelgewtr., Ø 16 x 4 mm, 1 Kugelgewtr., Ø 12 x 2 mm
- 3 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

„isel"-Schrittm.-Rundschalttisch 1 DM 568,00

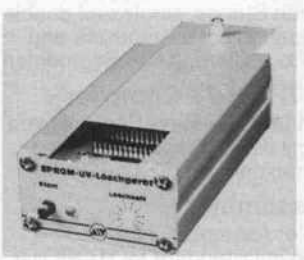
- Präzisions-Rundtisch, Höhe 90 mm, Gewicht 3 kg
- Alu-Gehäuse, L 250 x B 120 x H 60 mm, mit 2 Lagern
- Alu-Drehteller, Ø 240 mm, mit T-Nuten und Skalenring
- Spielfreier Zahnriemenantrieb, Untersetzung 1:4
- 1 Schrittmotor, Schrittwinkel 1,8°, Haltemoment 55 Ncm
- 1 End- bzw. Referenzschalter (Öffner), Genauigk. < 1/100 mm

Schrittmotor-Rundschalttisch 2 (o. Abb.) 1915,00

- Präzisions-Rundtisch, Höhe 80 mm, Gewicht 15 kg
- Gehäuse u. Grundplatte aus Grauguß, Drehteller aus Stahl
- Präzisions-Drehteller, Ø 200 mm, mit T-Nuten u. Skalenring
- Spielfrei einstellb. Schneckengetriebe, Untersetzung 1:72
- Zweiph.-Schrittmotor, Schrittw. 1,8 Grad, Haltem. 110 Ncm
- 1 End- bzw. Referenzschalter, Genauigkeit < 1/100 mm

„isel"-Eprom-UV-Löschgerät 1 DM 98,00

- Alu-Gehäuse, L 150 x B 375 x H 40 mm, mit Kontrollampe
- Alu-Deckel, L 150 x B 55 mm, mit Schiebeverschluß
- Löschschlitz, L 85 x B 15 mm, mit Auflageblech für Eproms
- UV-Löschlampe 4 W, Löschzeit ca. 20 Minuten
- Elektronischer Zeitschalter, max. 25 min, mit Start-Taster
- Intensive u. gleichzeitige UV-Löschung v. max. 5 Eproms

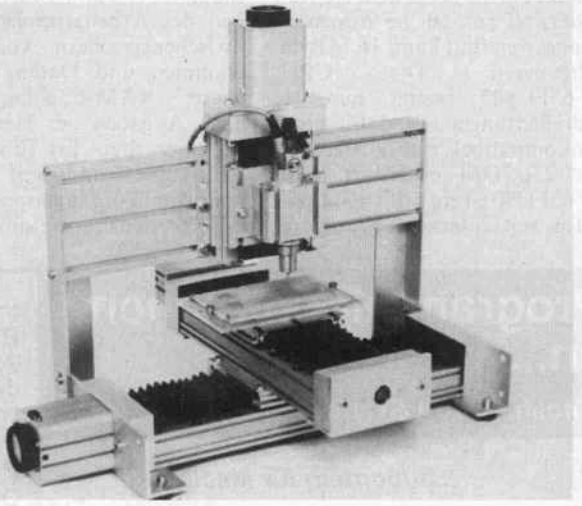


„isel"-Eprom-UV-Löscher. 2 (o. Abb.) DM 225,00

- Alu-Gehäuse, L 320 x B 220 x H 55 mm, mit Kontrollampe
- Alu-Deckel, L 320 x B 200 mm, mit Schiebeverschluß
- Vier Löschschlitze, L 220 x B 15 mm, mit Auflageblech
- Vier UV-Löschlampen, 8 W/220 V, mit Abschaltautomatik
- Elektronischer Zeitschalter, max. 25 Min., mit Start-Taster
- Intensive u. gleichzeitige UV-Löschung von max. 48 Eproms

„isel"-x/y/z-Anlage 1
3 Schrittmotoren u. 3 Kugelgewindetriebe 1995,00

- Präzis.-x/y-Kreuztisch m. 2 Vorschubeinh.
- Verfahrw. in x-Richt. 150 u. y-Richt. 250 mm
- Präzis.-z-Achse, Hub 80, m. Vorschubeinh.
- Alu-z-Balken, Länge 480 mm m. 2 Alu-Wink.
- 3 Schrittmot., Schrittw. 1,8°, Haltem. 55 Ncm
- 3 Schrittmotor-Abdeck., mit 9pol. Stecker
- 3 Kugelgew.-triebe, Ø 12 x 2 mm, m. 2 Lagern
- 3 End- bzw. Ref.-sch., Genauigk. < 1/100 mm



„isel"-x/y/z-Anlage 2 (ohne Abb.)
3 Schrittmotoren u. 3 Kugelgewindetriebe 2575,00

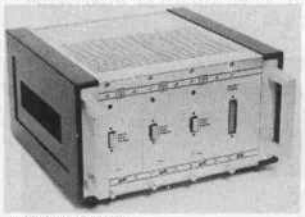
- Präzis.-x/y-Kreuztisch mit 2 Vorschubeinh.
- Verfahrw. in x-Richt. 250 u. y-Richt. 400 mm
- Präzis.-z-Achse, Hub 80, mit Vorschubeinh.
- Alu-z-Balken, Länge 760 mm, m. 2 Alu-Wink.
- 2 Schrittmot., Schrittw. 1,8°, Haltem. 110 Ncm
- 1 Schrittmot., Schrittw. 1,8°, Haltem. 55 Ncm
- 3 Kugelgew.-triebe, Ø 12 x 2 mm, m. 2 Lagern
- 3 End- bzw. Ref.-sch., Genauigk. < 1/100 mm

6419 Eiterfeld, ☎ (0 66 72) 70 31, Telex 4 93 150

Preise inkl. MwSt. Versand per NN, Katalog 3,– DM

„isel"-3-Achsen-Schrittmotorsteuerung DM 980,00

- 10-Zoll-Alu-Tischgehäuse mit vier 2-Zoll-Einschüben
- 1 Rückwandplatine mit 4 Steckverbindern nach DIN 41612
- 3 Steckkarten mit 2-Zoll-Frontplatte u. Netzteil 80 VA
- Bipolare Choppersteuerung, 45 V b. max. 2,5 A pro Phase
- Chopperfrequ. 27 kHz, max. 12000 Schritte (3600 U/min)
- Adapterkarte mit 2-Zoll-Frontplatte für User-Port-Eingang
- Steuer-Eing. fronts. über 25pol. Subminiatur-D-Stecker
- 3 Steuer-Ausg. fronts. über 9pol. Subminiatur-D-Stecker

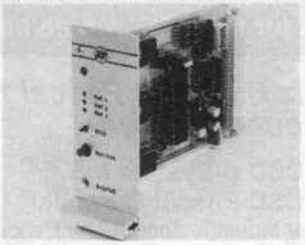


„isel"-Zweiphasen-Schrittmotorsteuerkarte (einzeln) DM 225,00

- Europa-Karte mit 2-Zoll-Frontplatte u. Netzteil 80 VA
- Bipolare Choppersteuerung, 45 V b. max. 2,5 A pro Phase
- Chopperfr. 27 kHz, max. 12000 Schritte/sec (3600 U/min)
- Voll- u. Halbschrittbetr., Stromabsenkung u. Ausschaltung
- Eingang für Richtung u. Schritt, einstellbarer Phasenstrom
- Steuer-Eing. über 16pol. Steckverbinder nach DIN 41612
- Steuer-Ausg. fronts. über 9pol. Subminiatur-D-Stecker

„isel"-Interface-Karte DM 565,00

- Euro-Karte mit Prozessor für max. 3 Achsen (Schritte)
- 2-Zoll-Frontplatte mit Bedienungs- und Anzeigeelementen
- Serielle Schnittstelle mit autom. Baudratenkennung
- Lineare Interpolation der angeschlossenen x/y-Achsen
- Programmierbare Beschleunigungs- und Bremsrampen
- Max. einstellbare Geschwindigkeit > 7000 Schritte/sek
- Steuerausgang rückw. über 64pol. Stecker, DIN 41612
- Serieller Eingang frontseitig über 3,5-mm-Klinkerbuchse



„isel"-µP-Relaiskarte (ohne Abb.) DM 450,00

- Euro-Karte mit 2-Zoll-Frontplatte u. 10-VA-Netzteil
- Serielle Schnittstelle mit autom. Baudratenkennung
- 7 programmierbare Relaisgänge mit LED-Kontrolle
- Ruhe oder Arbeitskontakt umschaltb., max. Belastung 2 A
- Interner Speicher 8 x 8 k Eprom + 8 x 8 k Ram
- Spannungsvers. 220 V/50 Hz rückw. über DIN 41612 C
- Relaisausgang fronts. über 15pol. Sub-D-Buchsenst.
- Serieller Eingang fronts. über 3,5-mm-Klinkerbuchse

isel-fotopositiv-beschichtetes Basismaterial

Pertinax FR 2, 1seitig, 1,5 mm stark, mit Lichtschutzfolie

Pertinax 100 x 160	DM 2,20	Pertinax 200 x 300	DM 8,30
Epoxyd FR 4, 1seitig, 1,5 mm stark, mit Lichtschutzfolie			
Epoxyd 100 x 160	DM 3,30	Epoxyd 200 x 300	DM 12,30
Epoxyd 160 x 233	DM 7,70	Epoxyd 300 x 400	DM 24,60
Epoxyd FR 4, 2seitig, 1,5 mm stark, mit Lichtschutzfolie			
Epoxyd 100 x 160	DM 4,50	Epoxyd 200 x 300	DM 16,75
Epoxyd 160 x 233	DM 10,50	Epoxyd 300 x 400	DM 33,50

ab 10 Stück 10%, 50 Stück 20%, 100 Stück 30% Rabatt

„isel"-UV-Belichtungsgerät 1 DM 198,00

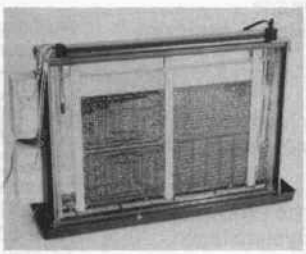
- Elox. Alu-Gehäuse, L 320 x B 220 x H 55 mm, mit Deckel
- Vier UV-Leuchtstoffrohren, 8 W/220 V, mit Reflektor
- Belichtungsfläche 245 x 165 mm, für Filme und Platten

„isel"-UV-Belichtungsgerät 2 DM 298,00

- Elox. Alu-Gehäuse, L 480 x B 320 x H 60 mm, mit Glasplatte
- Vier UV-Leuchtstoffrohren, 15 W/220 V, mit Reflektor
- Belichtungsfläche 365 x 235 mm, für Filme und Platten

„isel"-Entwickl.- u. Ätzgerät 1 (o. Abb.) DM 148,00

- Superschmale Glasküvette, H 290 x B 260 x T 30 mm
- PVC-Küvettenrahmen mit Kunststoffwanne
- Spezialpumpe, 220 V, mit Luftverteilrahmen
- Heizstab, 100 W/220 V, regelbar, Thermometer
- Platinenhalter, verstellbar, max. vier Euro-Karten
- Entwicklerschale, L 400 x B 150 x H 20 mm

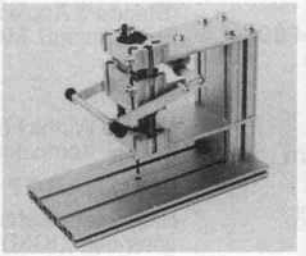


„isel"-Entwickl.- u. Ätzgerät 2 DM 225,00

- Superschmale Glasküvette, H 290 x B 430 x T 30 mm
- PVC-Küvettenrahmen mit Kunststoffwanne
- 2 Spezialpumpen mit Doppelluftverteilrahmen
- Heizstab 200 W/220 V, regelbar, Thermometer
- Platinenhalter, verstellbar, max. 8 Euro-Karten
- Entwicklerschale, L 500 x B 150 x H 20 mm

„isel"-Bohr- und Fräsgerät 1 DM 148,00

- Leistungsstarker Gleichstrommotor, 24 V, max. 2 A
- Spindel 2fach kugelgelagert, mit 1/8-Zoll-Spannzange
- Drehzahl 20000 U/min, Rundlaufgenauigkeit < 0,03 mm
- Präzisionshubvorrichtung mit 2 Stahlwellen, 8 mm Ø
- Verstellbarer Hub, max. 30 mm, mit Rückstellfeder
- Alu-T-Nutentisch, 250 x 125 mm, Arbeitsfeld 200 mm



„isel"-Bohr- u. Fräsgerät 2 (o. Abb.) DM 340,00

- Leistungsstarker Gleichstrommotor, 24 V, max. 2 A
- Spindel 2fach kugelgelagert, mit 1/8 Zoll-Spannzange
- Drehzahl 20000 U/min., Rundlaufgenauigkeit < 0,02 mm
- Linear-Vorschubeinheit, L 200 x B 125 x T 60 mm
- Präzisionshubvorrichtung mit „isel"-Linearführung
- Verstellbarer Hub, max. 80 mm, mit Rückstellfeder
- Alu-Gestell mit Alu-T-Nutentisch, 475 x 250 mm

„isel"-Aluminium-Gehäuse und Profile

1560	10-Zoll-Alu-Gehäuse (siehe Foto)	DM 67,80
1562	19-Zoll-Alu-Gehäuse, eloxiert	DM 99,80
1552	19-Zoll-Alu-Rahmen, eloxiert	DM 36,80
1573	1-Zoll-Frontplatte, 2 mm, eloxiert	DM 0,90
1575	2-Zoll-Frontplatte, 2 mm, eloxiert	DM 1,45
1591	Frontplattenschnellverschluß mit Griff	DM 0,85
1593	Frontplatte-Leiterplattenbefestigung	DM 0,70
1595	Führungsschiene (Kartenträger)	DM 0,55

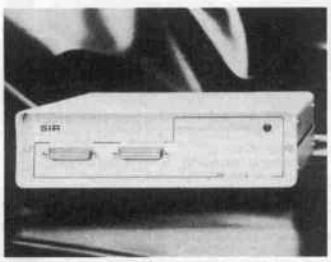
810	„isel"-Gehäuseprofil 1, eloxiert, Länge 1 m	DM 6,80
820	„isel"-Gehäuseprofil 2, eloxiert, Länge 1 m	DM 6,80
830	„isel"-Universalprofil 1, eloxiert, Länge 1 m	DM 7,80
850	„isel"-19-Zoll-Profil 1, eloxiert, Länge 1 m	DM 8,90

ab 10 Stck. 10%, 20 Stck. 20%, 50 Stck. 30% Rabatt

SIR testet

Für den Anschluß an ein Terminal ist der Seriendatentester 'SIR' von MicroLabor konzipiert, der es ermöglicht, serielle Daten nach V.24/RS232-Spezifikationen zu analysieren. Das Terminal dient als Ein-/Ausgabegerät, während es ein zweiter Anschluß erlaubt, sowohl DTE- als auch DCE-Geräte für Testzwecke mit dem SIR zu verbinden. Eine speziell entwickelte Programmiersprache für komplexe Testabläufe besitzt 55 Befehle. Das Gerät eignet sich zum Beispiel zur Baudratenmessung oder zur Bitmusterabtastung und ist für 4503 DM erhältlich.

Conitec, Grafenstraße 31, 6100 Darmstadt, 0 61 51/260 13.



Hardware-Spooler

Einen grafikfähigen Spooler mit Centronics-Schnittstelle gibt es von Lehner. Der 'HCPS-512' ist wahlweise mit Speicherkapazitäten von 128 K, 256 K oder 512 K lieferbar und kostet circa 800 DM. Ein spezielles Kompressionsverfahren ermöglicht optimale Speicherausnutzung und verkürzt das Drucken von Tabellen und Quelltexten.

H. Lehner, Opto- und Spezial-Elektronik GmbH, Dorfstraße 22, 2351 Wiemersdorf, 0 41 92/40 21 + 22

Neue 6502-CMOS-ICs

Inzwischen lieferbar sind drei neue CMOS-ICs für die 6502-Familie. Die 16-Bit-CPU W 65FC816 kostet 298 DM, ist aufwärtskompatibel zum 6502, besitzt auf 16 Bit erweiterte Register, elf zusätzliche Adressierungsarten und kann 16 MByte adressieren. Die CPU W 65FC802 besitzt nur 16 Adreßleitungen, ist dafür aber pinkompatibel zum 6502 und für 256 DM erhältlich. Der W 65FC90 ist ein 8-Bit-Komparator mit internem 'Compare Data Register', kostet 32 DM und eignet sich zum 'Soft Chip Select', für 'Soft-Breakpoints' oder als Bus-Monitor. Als 'Half Peripheral Interface Adaptor' wird der W 65FC29 bezeichnet, der acht einzeln richtungsprogrammierbare I/O-Leitungen aufweist, im Low-Zustand 8 mA treibt und für 23,50 DM zu haben ist.

Unitronic GmbH, Münsterstr. 338, 4000 Düsseldorf 30, 02 11/62 63 64-67.

Neuer Macintosh

Einen 1-MByte-Arbeitsspeicher und ein 800-KByte-Laufwerk hat der Macintosh Plus aufzuweisen, der sich damit zum schnellen Arbeiten mit integrierten Programmpaketen eignet. Der Rechner mit der 68000-CPU kann zudem einen Teil des Arbeitsspeichers zum Zwischenspeichern von Programmen und Daten nutzen. Dieses RAM-Caching führt nach Angaben des Herstellers zu einer drei- bis fünffachen Arbeitsgeschwindigkeit gegenüber dem Vorgängermodell, da es Diskettenzugriffe minimiert. Aufrüstbar ist der neue Macintosh mit einem zweiten, externen 800-KByte-Laufwerk und mit bis zu 4 MByte RAM. Wie beim Vorgänger läßt sich auch eine HyperDrive-Festplatte einbauen. Der Macintosh Plus kostet 9975 DM, sein Vorgänger kann für 3306 DM (Macintosh 512 KByte) beziehungsweise für 3876 DM (Macintosh 128 KByte) mit neuem ROM (128 KByte), neuem Laufwerk und neuer Hauptplatine aufgerüstet werden.

Apple Computer GmbH, Ingolstädter Str. 20, 8000 München 45, 0 89/3 50 34-0

Refresh Controller

Für 256K-DRAMs ist der Refresh Controller VL4502 von VTI besonders geeignet. Ohne zusätzliche Treiber adressiert er bis zu 2 MByte. Der Baustein ist in drei Geschwindigkeitsversionen für Speicher mit 150, 200 und 250 ns Zugriffszeit verfügbar. Im 48poligen DIL-Gehäuse kostet er 74 DM.

bitronic GmbH, Dingolfinger Str. 6, 8000 München 80, 0 89/41 80 07-0.

Wir haben die Programmiersprachen, die Sie brauchen...

... für MS-DOS, Apple Macintosh, ATARI ST, usw.

...z.B. UCSD Pascal mit neuen Preisen:

UCSD p-System mit UCSD Pascal-Compiler
Der Industriestandard für Pascal. Vollständiges p-System für IBM, ATARI ST und fast alle anderen gängigen Mikrocomputer. Mit voller Speicherausnutzung. Mit Bildschirmeditor und Filer.
DM 498,-*

Als **Zusatzprodukte** zum UCSD p-System lieferbar:

FORTRAN-77 Compiler
DM 398,-*

BASIC Compiler
DM 398,-*

Die Programmiersprachen können miteinander gemischt werden, auch in einem Programm.

... UCSD Pascal ist außerdem lieferbar unter:

UNIX, VMS (VAX), RT 11/TSX, RSX/RSTS

* Alle Preise sind empfohlene Endverbraucher-Preise incl. MwSt. • GEM ist ein Warenzeichen von Digital Research, Apple und Lisa von Apple, ATARI von ATARI Corp., Macintosh ein für Apple lizensiertes Warenzeichen.

...außerdem für Macintosh:

MacAdvantage: UCSD Pascal
Nutzt **alle** ROM-Routinen. Maclike Editor. Volle Speicherausnutzung.
DM 598,-*

MacAdvantage: Assembler
DM 498,-*

OnStage
Lisa Pascal auf dem Mac. Zugriff auf **alle** Toolbox-, Betriebssystem- und AppleTalk-Routinen. (Demnächst lieferbar.)

Consulair Mac C
Mit Unterstützung für SANE (Standard **A**pple **N**umeric **E**nvironment). Mit neuem Consulair Linker.

Modula 2
Niklaus Wirths Fortschreibung von Pascal. Kompakter, schneller Code.

FORTRAN-77
Naturwissenschaftliche Probleme unter dem UCSD p-System lösen.

(Auf der CeBIT '86 finden Sie uns auf dem Apple-Stand in Halle 13)

...außerdem für ATARI ST:

Pro Pascal Compiler
Zwei-Pass Compiler von Prospero mit dynamischen Strings, single und double precision floating point Arithmetik und separater Compilierung von Segmenten.

Pro FORTRAN-77 Compiler
Optimierender Zwei-Pass Compiler von Prospero mit IEEE-Standard Floating-Point-Format, nach ANSI X3.9-1978.

Makro-Assembler
Komfortabler Assembler von Metacomco mit Standard MOTOROLA Mnemonics, Linker und Bildschirmeditor.

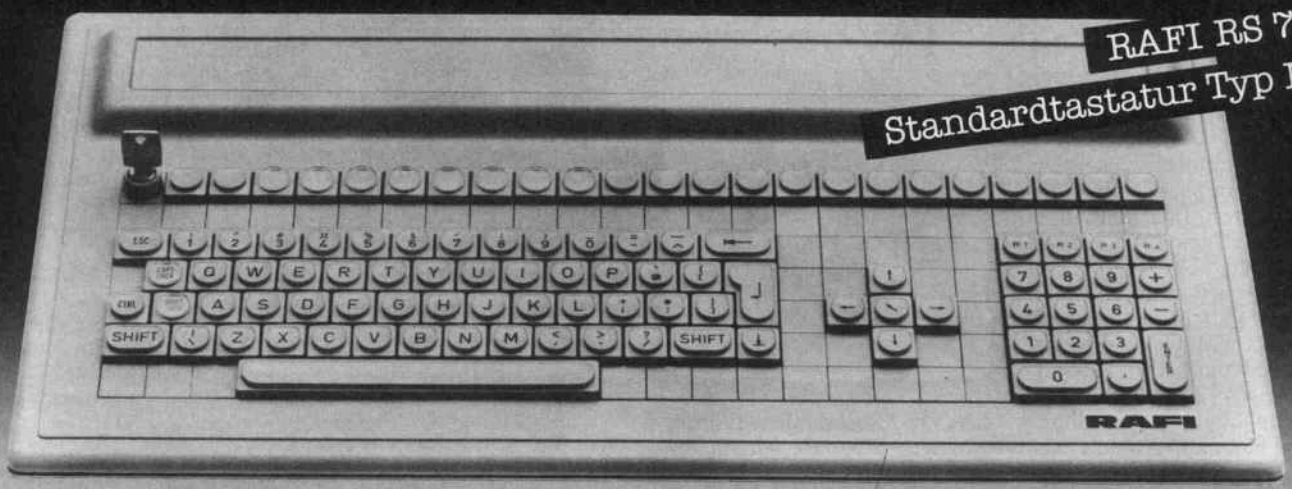
Modula-2 Compiler
Von TDI modulare GEM-Programmierung in der neuen Sprache von Niklaus Wirth. Der Compiler erzeugt Maschinencode. Bildschirmeditor in GEM-Technik.

TLDU
The **L**ast **D**isk **U**tility
Jetzt auch für ATARI ST.

(Auf der CeBIT '86 finden Sie uns auch auf dem ATARI-Stand.)

FOCUS Computer GmbH

Friesenstraße 14
D-3000 Hannover 1
Tel. 0511-345461
Teletex 5118476=FOCUS
Generalvertretung Schweiz: ILR AG
Universitätsstraße 87
CH-8033 Zürich



Komfort-Dateneingabe. Flexibel und freiprogrammierbar.

Aus der bewährten Tasterbaureihe RS 76 wurde diese neue Standardtastatur entwickelt. Mit einer fast nicht zu überbietender Flexibilität im Bereich Datenerfassung bzw. der Textverarbeitung. Das gesamte Tastaturenfeld ist alphanumerisch nach DIN 2137. Mit numerischem Eingabe- und Cursorblock. In der 7. Tastenreihe 23 Funktionstasten, ein Schloßschalter. Codiert nach ASC II (DIN 66003). N-Key-Rollover Verriegelung.

Betriebsspannung +5 Volt. Einbautaster wahlweise in kontaktloser (Hall-IC) oder mechanischer Ausführung. Datenausgabe über eine asynchrone vollduplexe Seriellschnittstelle. Übertragungsrate 1200 Baud über Diodenschaltung 300–9600 Baud variabel. Steckverbindungen und Adapter ermöglichen V24 und TTY-20 mA Schnittstellen. Alle codierten Tasten sind freiprogrammierbar, auch mit anderen Codes und Strings.

RAFI GmbH & Co. · Elektrotechnische Spezialfabrik
Postfach 2060 · D-7980 Ravensburg
Telefon (07 51) *89-0 · Telefax (07 51) 89 300 · Telex 732866 · Btx *4 56 56#

Wir stellen aus: CeBIT Hannover, Halle 17, Stand 406

Commodore

PREISSENKUNG!
COMMODORE PC 10, 256 K RAM, 2 Floppys à 360 K incl. Monitor, nur noch ..... 3769,—
COMMODORE PC 20 nur noch ... 5769,—
COMMODORE C 128 Home-Computer a. A.
Voraussichtlich in Kürze lieferbar:
COMMODORE AMIGA (deutsche Ausführung).

TANDON

NEU: TANDON PC, 256 K, CPU 8088, IBM-PC-kompatibel, incl. 14" Monochrom-Monitor, dt. Tastatur, MS-DOS 2.11 und GW-BASIC mit 2 Floppys à 360 K .... 3848,—
PCX 10, 10 MB Platte, 1 Floppy .. 4998,—
NEU: TANDON PCA, 512 K RAM, CPU 80286, IBM-AT-kompatibel, 1 Floppy 1,2 MB incl. 14" Monochrom-Monitor, dt. Tastatur, MS-DOS 3 und GW-BASIC PCA 20, mit 20 MB Platte ..... 8489,—

Schneider COMPUTER DIVISION

SCHNEIDER CPC 464, CPC 664 und CPC 6128 und JOYCE PCW 8256 auf Anfrage.

PHILIPS

PHILIPS PC YES: ein oder zwei 3½" Floppys à 720 K, incl. DOS +, BASIC, OPEN ACCESS usw. ..... a. A.
PHILIPS P 2000 ..... ab 2645,—

PLANTRON

PREISSENKUNG!
PLANTRON PT-16 XT, IBM-kompatibel, 256 K RAM, 2 Floppys à 360 K, Color-Grafikkarte incl. MS-DOS ..... nur 3195,—
NEU: PLANTRON PT 16 AT/20, IBM-AT-kompatibel, 640 K RAM, ein Floppy 1,2 MByte und 25,6 MByte Platte .... 7789,—

MULTITECH

MULTITECH PC-POPULAR 128 K RAM, CPU 8088, IBM-kompatibel, 1 Floppy 360 K, dt. Tastatur, Color-Grafikkarte, incl. MS-DOS 2.11 und Monitor ..... 2445,—
dto. wie oben, jedoch mit 80-Zeichen-Farbmonitor ..... 3089,—
Zweites Floppy 360 K..... 478,—

apricot

PREISSENKUNG:
APRICOT F1 STUDENT, 256 K RAM, CPU 8086, dt. Tastatur, ein 3½" Diskettenlaufwerk 711 K incl. 12" Monochrom-Monitor, MS-DOS 2.11 und div. Software nur 2798,—
NEU: APRICOT F2 512 K RAM, CPU 8086, dt. Tastatur, mit zwei 3½" Diskettenlaufwerken 711 K, 9" Monochrom-Monitor, Infrarot-Mouse, MS-DOS 2.11, GEM und div. Software (voraussichtlich in Kürze lieferbar) ..... 4989,—
PREISSENKUNG bei APRICOT F10 und APRICOT PC/XI. Preise auf Anfrage.

ATARI

Die neuen ATARI-Computer weit unter den empfohlenen Verkaufspreisen von ATARI.

brother

BROTHER HR-15XL II Typenraddr. 1195,—
BROTHER M 1509 Matrix-Drucker . 1395,—
BROTHER Twinriter 5 ..... 3658,—

OKIDATA

OKI Microline 182 Drucker..... 765,—
OKI Microline 183 Drucker..... 945,—
OKI Microline 192 Drucker..... 1225,—
OKI Microline 193 Drucker..... 1570,—

JUKI

JUKI 6000 Typenraddrucker ..... 690,—
JUKI 6100 Typenraddrucker ..... 1198,—
JUKI 2200 P/S Schreibmaschine mit Interface ..... 885,—
JUKI 5520 Farb-Matrix-Drucker ... 1385,—

Panasonic

PANASONIC RL-H 7000 IBM-kompatibel, 256 K RAM, 2 Floppys à 360 K ..... a. A.
KX-P1080 Matrix-Drucker ..... a. A.
KX-P1092 Matrix-Drucker ..... a. A.
KX-P1592 Matrix-Drucker ..... a. A.

star

STAR SG 10 Matrix-Drucker ..... 789,—
STAR SG 10/C Matrix-Drucker.... 739,—
STAR SG 15 Matrix-Drucker ..... 1135,—
STAR SD 10 Matrix-Drucker ..... 1075,—
STAR SD 15 Matrix-Drucker ..... 1439,—
STAR SR 10 Matrix-Drucker..... 1495,—
STAR SR 15 Matrix-Drucker..... 1789,—
NEU: STAR NL 10 a. A.
Preise mit engl. Handbuch, deutsche Handbücher DM 26,—/St. Alle Drucker außer SG 10/C für IBM-PC umschaltbar.

C. ITOH

RITEMAN F+ Matrix-Drucker..... 795,—
RITEMAN C+ Matrix-Drucker .... 725,—
ITOH 8510B Matrix Drucker (NLQ) 1260,—

TAXAN

TAXAN-Monitore zu interessanten Preisen.

EPSON

EPSON LX 80 Matrix-Drucker .... 780,—
EPSON FX 85 Matrix-Drucker .... 1285,—
EPSON FX 105 Matrix-Drucker ... 1715,—
EPSON LQ 1500 Matrix-Drucker .. 3190,—

Wir führen außerdem: TRIUMPH-ADLER, SPECTRAVIDEO, SANYO, GENIE, OLIVETTI, FUJITSU, CANON, COMPUDATA, TOSHIBA, HEWLETT PACKARD, Maxell usw.

Fordern Sie bitte kostenlos die aktuelle Preisliste über unser gesamtes Lieferprogramm an, oder besuchen Sie uns. Bei Anforderung von Prospektmaterial bitte Produkt angeben. Wir haben Montag bis Freitag von 9.00 bis 13.00 Uhr und 14.30 bis 18.00 Uhr geöffnet. **Selbstverständlich können Sie auch telefonisch bestellen.** Preise in DM incl. 14% gesetzlicher Mehrwertsteuer zuzügl. Versandselbstkosten. Versand per Nachnahme. Auf sämtliche Geräte haben Sie 6 Monate Garantie. Alle Preise beziehen sich auf den vollen Lieferumfang, wie vom Hersteller angeboten. Das Angebot ist freibleibend. Wir bitten um Beachtung, daß bei großer Nachfrage nicht immer jeder Artikel sofort lieferbar ist.

MICROCOMPUTER-VERSAND

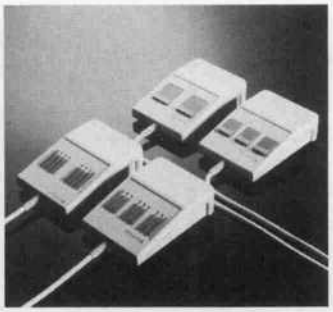
ernst mathes GmbH

Pohlstraße 28, 4419 Laer, Tel.: 02554/1059

NEUE ANSCHRIFT!

Genügsame CMOS-Maus

Logitech kündigt eine neue CMOS-Maus an, die ihren Versorgungsstrom (maximal 5mA) über die RTS-und DTR-Leitungen bezieht. Der Signalpegel kann zwischen 6 und 15 Volt liegen, wodurch sie praktisch auf jedem System eingesetzt werden kann. Die 'LOGI-MOUSE C7' bietet eine Auflösung von 8 Punkten je Millimeter und kann auf Wunsch auch mit 13 Punkten je Millimeter geliefert werden. Es stehen Modelle mit zwei oder drei Tasten zur Auswahl, die verschiedene Anschlußoptionen zu bieten haben. Die Mäuse laufen mit jeder gängigen Software, einschließlich AutoCAD, Lotus 1-2-3, MSWord und GEM.



Jean-Luc Mazzone und Martine Cochet, Logitech SA, CH-1143 Apples Schweiz, 0041/21 774 545

Computer-Gestell

Für Computerbesitzer, die viele Kabel am Arbeitsplatz nicht mögen, bietet die Firma Tonacord für 175 DM ein Tischgestell an, in dem sich der Atari 260/520 oder auch der C128 von Commodore unterbringen läßt.

Tonacord Tontechnik, Postfach 1444, 2330 Eckernförde, 0 43 51/4 11 22/ 40 39

Enterprise jetzt für 798 Mark

Die Enterprise Computers GmbH hat die empfohlenen Verkaufspreise für den Rechner Enterprise 128K und Zubehör (Test in c't 3/86) beträchtlich gesenkt: Statt 1198 DM kostet das Grundgerät jetzt nur noch 798 DM. Das Diskettenlaufwerk inklusive Controller ist für 998 DM zu haben.

DC/DC-Wandler

Durch Eingangstoleranzen von ±25% zeichnet sich der DC/DC-Konverter 'SI 10 W' aus, der sich besonders zum Aufbau batteriegepufferter Versorgungen eignet. Das Gerät mit den Abmessungen 51 x 51 x 10 mm liefert bei Eingangsspannungen von 12V, 24V, 48V oder 60V eine Ausgangsleistung von 10 Watt, wie zum Beispiel 5V/2A. Die Wandler arbeiten mit Pulsbreitenregelung, sind damit dauerkurzschlußfest und haben Wirkungsgrade von 80% bis 90%. Der Preis für diese Wandler beträgt 181 DM.

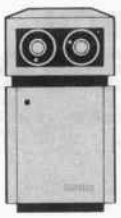
Brandner Vertriebs-GmbH, Stresemannstr. 19, 6450 Hanau 1, 0 61 81/ 3 20 37.

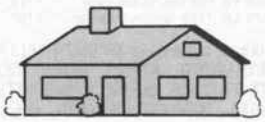
Buchprogramm stark erweitert

Die Verlagsszene für Computer-Fachliteratur verändert sich: Im Februar 1986 übernahm der Verlag Heinz Heise GmbH, Hannover, Computerfreaks und -profis vor allem bekannt durch 'c't – Magazin für Computertechnik', 'elrad – Magazin für Elektronik', 'INPUT64 – das elektronische Magazin' und einige erfolgreiche Sachbücher, den Verlag W.D. Luther mit seinem umfangreichen Computer-Buchprogramm. W.D. Luther wird zukünftig für den Heise-Verlag und von Hannover aus mit Autoren verhandeln, Manuskripte redigieren und das nun recht respektable Buch- und Software-Programm des Heise-Verlags zügig erweitern.

An alle, die mit ihrem „PC" telefonieren wollen

Mit ASCOM IV schaffen Sie Verbindungen:

Zur ON LINE Datenbank  oder zu Datex P der DBP, auch zu Micro-, Mini- oder Mainframe Computern, selbst von zu Hause

 zum Computer im Büro.

ASCOM IV läßt sich unmittelbar ohne Probleme in Ihr vorhandenes PC-System integrieren. Leicht verständliche Menüs zeigen die Auswahlmöglichkeiten auf dem Bildschirm an. Der eingebaute Editor ermöglicht die Zusammenstellung und Bearbeitung der zu übertragenden Texte und Informationen. Durch das mitgelieferte Programm „schwarzes Brett" können Ihre Kunden und Geschäftsfreunde zu jeder Zeit Informationen abrufen bzw. Nachrichten hinterlassen. ASCOM IV läuft unter PC-DOS und MS-DOS auf IBM PC und kompatiblen Systemen.

Bestell-Nr. 111170 **DM 936,20**
Versand per Nachnahme incl. MwSt.

ASCOM IV, PC-DOS, MS-DOS und IBM PC sind eingetragene Warenzeichen.

Wir informieren Sie gerne. **Wählen Sie: 07841/4500**

Händleranfragen willkommen.

PFOTENHAUER Microcomputer-Anwendung
Postfach 1312 · Neulandstraße 16 · 7590 Achern · Telex 752100 msoft

pfotenhauer
MICROCOMPUTER-ANWENDUNGEN

top kit electronic

Der **HERSTELLER** professioneller Monitore vom Profi in
3″ - 5″ - 7″ - 8″ - 9″ - 12″ - 14″ - 15″

für alle Anwendungen in der
★ DATENVERARBEITUNG als Text-, Graphiksysteme und für Maschinensteuerung
★ VIDEOTECHNIK als Überwachungsanlage, für Bilddarstellung, für medizinische Geräte

mit folgenden Eigenschaften:
- ○ geprüfte Qualität MADE IN GERMANY, erhältlich in Monochrom und Farbe - 10″ - 15″ -
- ○ alle Monitore mit/ohne intelligenter Terminal-Karte.
- ○ extrem ruhiger Bildstand.
- ○ kompatible Anschlußschnittstelle.
- ○ kundenspezifisch in der Electronic oder Mechanik — einsatzbezogen —, z. B. IP 65 oder IP 54
- ○ 19″ Technologie. Einschübe — Cassetten für - 3″ - 5″ - 7″ - 8″ - 9″ - 12″ Monitore.
- ○ 5 V DC Ausführung für Labor oder mobilen Einsatz.
- ○ Dauerbetriebssicher.
- ○ 100 Stunden Sicherheitstest.
- ○ 1 Jahr Garantie.
- ○ über 50 verschiedene Gehäuse-Variationen — eigener Formenbau.

top kit electronic

Der **HERSTELLER** und **LIEFERANT** mit kundenbezogener Kundenberatung und höchstem Qualitätsstandard. Ein Prüfzertifikat liegt allen unseren Produkten bei.

top kit electronic Vertriebs GmbH, Breslauer Weg 33, 8192 Geretsried 2
Telefon (0 81 71) 3 10 28-29, Telex 5 27 895 tke d

HEISE

M. B. Immerzeel

MSX-BASIC programmieren lernen

Über 60 Programme, 20 Abbildungen, 7 Tabellen und 12 Bildschirmfotos

Dieses Buch beschreibt die Programmiersprache Microsoft Extended BASIC, die bei allen Computern der MSX BASIC-Version 1.0 verwendet wird.

Die Funktionsweise der einzelnen Befehle wird anhand von Beispielprogrammen ausführlich erklärt. Durch die Eingabe dieser Programme, dem Ausprobieren verschiedener Möglichkeiten, dem Überprüfen der Ergebnisse auf dem Bildschirm und der Erklärungen in diesem Buch werden die Programmierfertigkeiten des Lesers Schritt für Schritt erweitert, so daß er sehr bald in der Lage sein wird, eigene Programme zu schreiben. In diesem Zusammenhang werden auch Aufbau und Arbeitsweise des Computers anschaulich beschrieben.

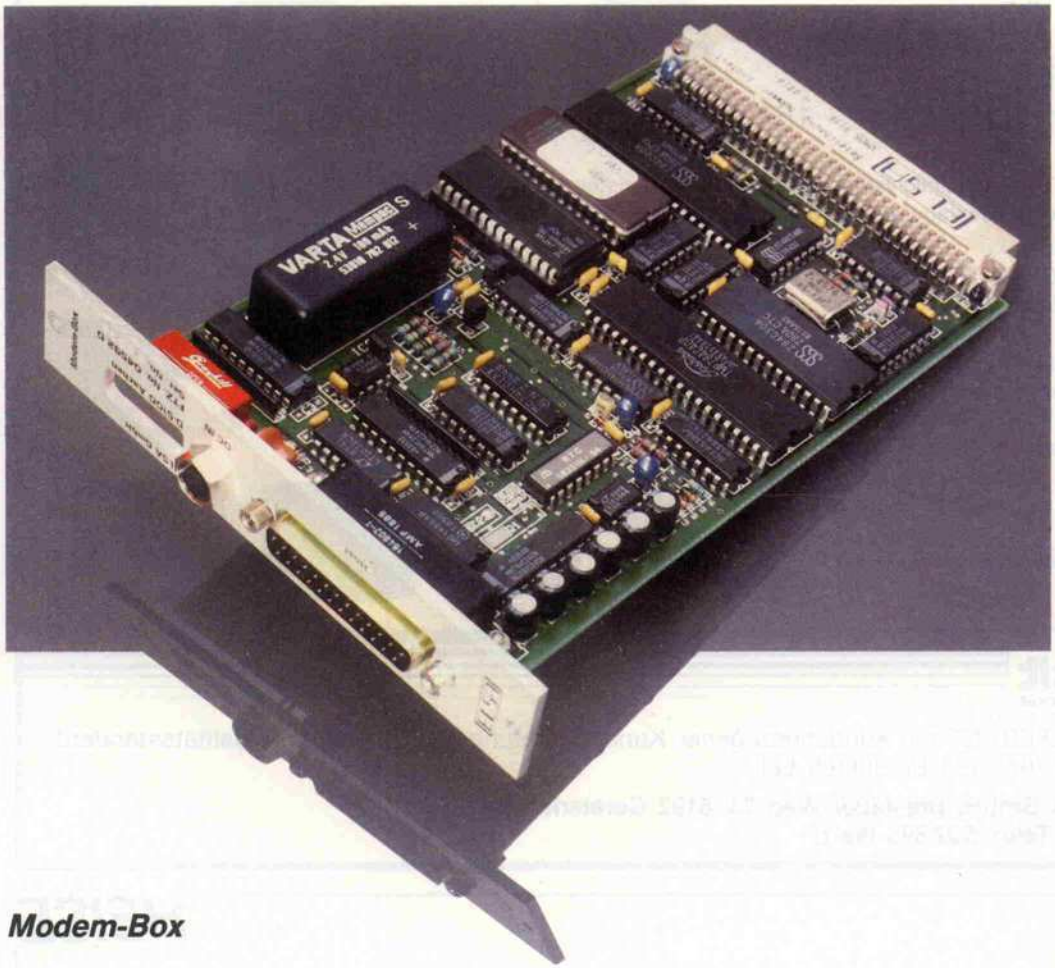
Dieses Buch richtet sich in erster Linie an Computer-Laien, doch werden auch die schon etwas Fortgeschritteneren so manche interessante Anwendungsmöglichkeit darin finden, da der Schwerpunkt des Buches auf dem Tongenerator und den grafischen Möglichkeiten des Computers liegt.

Inhalt: Aufbau und Bedienung des Computers, Befehle, Variablen, Programmaufbau, Direct- und Program-Mode, ASCII-Code, Binär-, Oktal- und Hexadezimalzahlen, Zufallszahlen, Timer, Tongenerator, Grafische Funktionen.

ca. DM 26,–
ca. 140 Seiten, Broschur
Format 16,4 x 22,9 cm

ISBN 3-922 705-29-4

Verlag Heinz HEISE GmbH · Postfach 61 04 07 · 3000 Hannover 61



Modem-Box

Schlüssel zur Welt

Manfred Spitzer

Der Horizont des eigenen PCs beschränkt sich auf RAM und ROM, Platten und Disketten. Um aus der engen Welt der eigenen Soft- und Hardware auszubrechen, wäre die Kommunikation mit anderen Rechnern wünschenswert: Der Anschluß an externe Datensysteme erschließt unerschöpfliche und stets aktuelle Informationsquellen. Ganz legal und mit dem Segen der Post kann man mittels der Modem-Box der Firma Elsa und einem gewöhnlichen Computer diese Quellen sprudeln lassen.

Da die ganze Welt ohnehin über das Telefonnetz verkabelt ist, bedarf es nicht unbedingt eines neuen Datennetzes zur Kommunikation von Computer zu Computer. Das Telefonnetz überträgt einen Frequenzbereich von 300 Hz bis 3,7 kHz, was dem Hauptanteil des Spektrums der menschlichen Sprache entspricht. Um die Nullen und Einsen der 'Computersprache' dem Telefonnetz verständlich zu machen, müssen sie umgewandelt werden, zum Beispiel in hohe und niedrige Frequenzen, also Töne. Diese Umwandlung besorgt ein Modem (MOdulator/DEModulator).

Akustikkoppler

Der einfachste und sicher auch kostengünstigste Anschluß des eigenen Computers an die große, weite Welt ist ein Modem auf der Basis eines Akustikkopplers, also die Übertragung von Schall auf die Hör- und Sprechmuschel des Telefonapparates mit Mikrofon und Lautsprecher. Somit ist ein Akustikkoppler nicht elektrisch mit dem Postnetz verbunden; trotzdem muß er von der Post zugelassen sein.

Eine direkte elektrische Verbindung mit dem Postnetz bietet aber verschiedene Vorteile: Die Übertragungsrate ist viermal so hoch wie beim Akustikkoppler, und die Übertragung ohne den Umweg über die Schallkopplung ist störsicher. Nicht zuletzt ist es auch dem Akustikkoppler nicht möglich, 'den Hörer selbst abzuheben', wenn es klingelt – wohl aber einem direkt gekoppelten Modem. Damit ist auch automatisches Wählen oder sogar das Betätigen der Erd-Taste bei Nebenstellenanlagen möglich. Diese direkte Kopplung eines Computers an das Telefonnetz unterliegt aber strengen Bedingungen.

Einbau-Modem

Etwa seit Mitte 1984 bietet die Post einen 'Modem'-Service: Für 20 DM im Monat kann man ein Einbau-Modem von der Post mieten, das direkt mit dem Netz gekoppelt wird. Das Einbau-Modem mit der Bezeichnung MDB 1200 hat Europakartenformat (160 × 100 mm) und wird in den Computer eingesteckt, der natürlich dafür die Genehmigung der Post haben muß. Diese Installation gehört bereits zur 'Intimsphäre' der Post: sie übernimmt selbst den Anschluß des Modems an das Telefonnetz und an den Computer. Der Rechner muß jedoch neben der üblichen FTZ-Nummer noch die Zulassung der Post für den Betrieb mit diesem Modem haben.

Modem-Box

Die Modem-Box der Firma Elsa ist ein Mikrocomputersystem, in das die Modem-Baugruppe MDB 1200-03 der deutschen Bundespost integriert wird.

Der Steuerrechner der Box ist ein 84C00P (Z80 CMOS-Version) mit einem Systemtakt von 2,4576 MHz, 32 KByte ROM, 2 KByte RAM und einem Host-Interface mit einstellbaren Baudraten von 300 bis 19200 Baud. Das Ganze sitzt auf einer ordentlich aufgebauten Epoxy-Platine im Europaformat. Das Post-Modem und der Steuerrechner sind in einem formschönen kleinen Einschubgehäuse untergebracht, das ansonsten lediglich einen Ein/Aus-Schalter und eine Leuchtdiode enthält. Die Stromversorgung erfolgt über ein separates kleines Netzgerät.

Nach dem Anschluß kann die Modem-Box selbsttätig Verbindungen aufbauen und Datentransfer durchführen. Dabei sind folgende Übertragungsarten möglich: 300/300 Baud vollduplex, 1200/75 Baud vollduplex, 75/1200 Baud vollduplex und 1200 Baud halbduplex. Der Zugriff auf das öffentliche Datennetz mit Paket-Vermittlung (Datex-P) über das Fernsprechnetz ist ebenfalls möglich.

Die Steuerung der Modem-Box erfolgt über ein beliebiges Computersystem, das lediglich eine serielle Schnittstelle haben muß,

über die es mit der Box verbunden werden kann. Alternativ zu einem Rechner kann man an die Modem-Box ein Bildschirmterminal anschließen, wovon einige Modelle von verschiedenen Herstellern mit in die Zulassung der Box aufgenommen sind.

Die Modem-Box kann verschiedene 'teure' kommerzielle Modems emulieren, wie zum Beispiel D300 oder D1200. Gängige Modem-Typen (Hayes Smart Modems oder Popcom X 100), wie sie beispielsweise in den Installations-Menüs von Sidekick oder Symphony auftauchen, werden jedoch nicht erwähnt.

Man kann die Modem-Box zusätzlich mit einem wartungsfreien Akku ausstatten, wodurch ein unterbrechungsfreier Betrieb unabhängig von der Netzversorgung möglich ist.

Betrieb

Nachdem man eine Vielzahl von DIL-Schaltern für die verschiedenen Betriebsarten und Übertragungsparameter zwischen Postnetz und Modem-Box sowie zwischen Modem-Box und Computer überprüft hat, kann man das System an der Frontplatte einschalten. Eine rote Leuchtdiode blinkt ein paarmal (das bedeutet Selbsttest und Modem-Test) und leuchtet dann kontinuierlich, was bedeutet: es ist alles in Ordnung. Würde bei der Diagnose ein Fehler auftreten, 'morst' die Leuchtdiode eine Sequenz, aus der man auf den Fehler schließen kann.

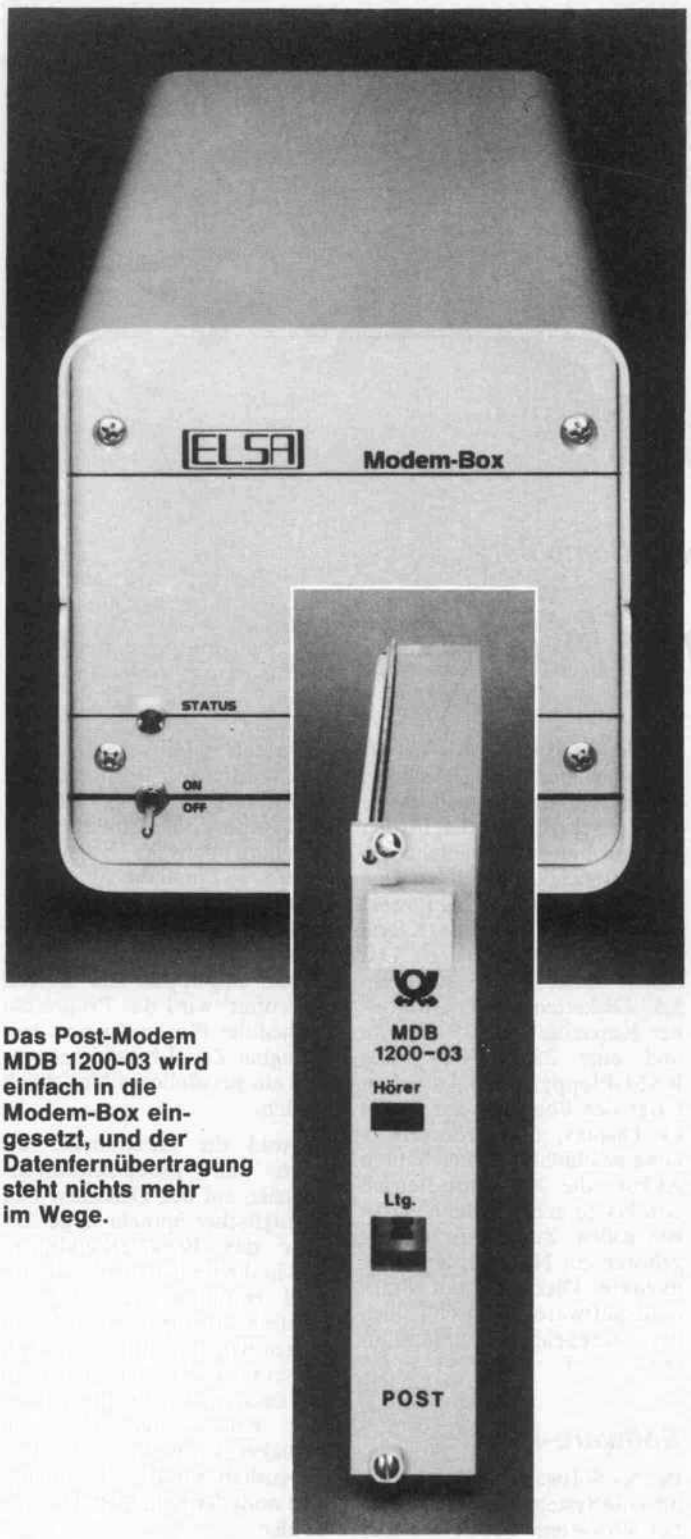
Smart-Modus

Im 'Smart-Modus' werden außer den Pins 1 und 7 (Erde, Masse) nur die Pins 2 und 3 der Schnittstelle zum Host-Rechner verwendet. Das bedeutet, daß die Steuerleitungen nicht unterstützt und auch nicht abgefragt werden. In dieser Betriebsart hat die Modem-Box keine Möglichkeit festzustellen, ob der Eingabepuffer des Host-Rechners bereit ist, Zeichen zu empfangen. Erhält die Modem-Box ein Zeichen über die Telefonleitung, sendet sie es ohne Software-Handshake umgehend an den Host-Rechner. Interne Modem-Funktionen werden durch 'Escape' gekennzeichnet.

PC-Modus

Im PC-Modus werden die Steuerleitungen abgefragt und bedient. Das bedeutet hinsichtlich der zu übertragenden Zeichen, daß die Modem-Box am Zustand der Statusleitungen unterscheiden kann, ob ein 'Escape' ein zu übertragendes Byte oder ein Steuerzeichen ist. Man kann also alle Zeichen senden und empfangen, ohne daß die Übertragung sich 'aufhängt'.

Es gibt verschiedene Programme zur Datenfernübertragung mittels Modem. Dem Testgerät lagen eine Diskette und ein Handbuch von 'CROSSTALK' der Firma Microstuff für MS-DOS-Rechner bei. Die Firma Elsa hat für dieses Programm eine Anpassung an die Modem-Box vorgenommen. Bis auf einen ungeklärten 'Ausstieg' (die ROM-residente Software meldete sich zwischendurch unaufgefordert) funktionierte sie auch gut.



Das Post-Modem MDB 1200-03 wird einfach in die Modem-Box eingesetzt, und der Datenfernübertragung steht nichts mehr im Wege.

Terminal-Modus

Schließt man an die Modem-Box ein Bildschirm-Terminal an (oder einen Computer mit einem Terminal-Emulationsprogramm), meldet sich die ROM-residente Sotware der Modem-Box mit einem kleinen Funktionsauswahl-Menü. Man kann die Übertragungsparameter ändern, einen ankommenden Anruf abwarten oder selbst wählen. Meldungen, wie 'Die Modem-Box wartet auf einen Anruf' oder 'Die Modem-Box wählt' oder 'Die Verbindung wurde abgebrochen', geben über Erfolg oder Mißerfolg der jeweiligen Operation auf dem Bildschirm Auskunft.

Problem FTZ-Nummer

Im Handbuch zur Modem-Box heißt es: 'Die Modem-Box übernimmt sämtliche Steuervorgänge des Modems und garantiert damit zuverlässig die Einhaltung der postalischen Vorschriften. Daher benötigt der Host-Rechner selbst keine spezielle FTZ-Zulassung für dieses Modem, es genügt eine allgemeine Zulassung.'

Das klingt sehr gut, und große PC-Hersteller wie IBM, Commodore oder Siemens haben selbstverständlich eine FTZ-Zulassung für ihre Geräte erwirkt. Jedoch für die vielen Besitzer von Taiwan-PC-Nachbauten dürfte hier der Traum von der großen weiten Welt am Ende sein – denn kaum ein Taiwan-PC hat eine FTZ-Nummer. Für all die Besitzer eines 'postgeprüften' Computers bietet die Modem-Box jedoch die Möglichkeit, ohne Probleme mit der Post ein direktgekoppeltes Modem zu betreiben. Die Modem-Box ist für etwa 1800 DM bei der Firma Elsa erhältlich.

Bezugsquelle:

Elsa
Gesellschaft für elektronische Systeme mbH
Monheimsallee 53
5100 Aachen
Telefon: 02 41/2 99 92

Bondwell Model 2 Portable-Computer

CP/M im Koffer

Martin Ernst

Bondwell stellt seinen CP/M-80-Computer Model 2 als den idealen Begleiter dar: Der Rechner ist leicht transportabel, bietet Hilfe beim täglichen Papierkrieg sowie bei Entscheidungen und befreit schließlich auch den Schreibtisch von überflüssigem Papier. Ist der Bondwell 2 also der ideale Computer, ein Alleskönner?

Für den Preis von 2385 DM erhält man einen CP/M-Rechner, der wirklich nur so groß ist wie die Zeitschrift, die Sie in den Händen halten – zugeschlagen versteht sich. Als CPU ist eine Z80 L (L steht für Low Power) eingebaut, die 64 KByte Hauptspeicher adressiert. Daneben bietet das System ein 3,5"-Diskettenlaufwerk mit einer Kapazität von 350 KByte und eine 256 KByte große RAM-Floppy. Seine Ausgaben tätigt der Bondwell auf einem LC-Display, die Stromversorgung geschieht aus eingebauten Akkus, die Non-Stop-Betrieb von bis zu acht Stunden erlauben sollen. Zum Lieferumfang gehören ein Netzadapter sowie mehrere Disketten mit Standard-Software einschließlich der dazugehörigen Handbücher.

Software-Pakete

Bei der Software handelt es sich um eine Systemdiskette mit dem Betriebssystem CP/M 2.2 und mehreren Hilfsprogrammen, den Micro-Pro-Programmen WordStar, CalcStar, DataStar und ReportStar. Außerdem erhält man noch eine Diskette mit einem Finanzbuchhaltungs-Programm und einer Adreßverwaltung, die sich aber als ein Beispielprogramm von Turbo-Pascal entpuppte. Als 'Special free offer' wird das Programm 'Scheduler Plus' auf einem beigefügten Zettel bezeichnet – es soll ein persönliches Notizbuch bilden.

Sowohl die Handbücher als auch die Dokumentations-Dateien auf den Disketten sind in englischer Sprache abgefaßt. Nur das Benutzerhandbuch, das in die Bedienung des Bondwell einführt, liegt in einer deutschsprachigen Ausgabe in einem Ringbuchordner vor. Zu dieser Grundausstattung hatten wir noch ein 'Ergänzungspaket', das eine Tragetasche, ein Drucker- sowie ein RS-232-Kabel enthält. Für diese Ergänzungen muß der Kunde 85 DM bezahlen.

Der Start des Computers brachte keine Probleme mit sich: LC-Display hochklappen, Ein/Aus-Schalter betätigen.

Erste Begegnung

Nach kurzer Zeit meldet dann der Computer, daß er sich überprüft hat und man durch Drücken einer beliebigen Taste in das Monitor-ROM gelangt. Anschließend muß man noch drei Fragen beantworten: ob das Betriebssystem von der RAM-Floppy oder der Diskette geladen werden soll, welche nationale Tastaturbelegung man wünscht und wie die Zuordnung der Diskettenstation und der RAM-Floppy zu den logischen Laufwerksbezeichnungen von CP/M erfolgen soll.

Diese Prozedur muß man jedesmal nach dem Einschalten des Computers durchlaufen. Warum spendierte Bondwell dem Modell 2 für diese Daten nicht batteriegepuffertes RAM? Mit einem Hilfsprogramm könnte man dann bei Bedarf die Daten ändern. Wünschenswert wäre auch Batterie- oder Akkupufferung bei der RAM-Floppy. So muß man nach jedem Einschalten des Rechners Dateien und Programme von Diskette ins RAM laden – eine recht umständliche Prozedur. Außerdem kommt es manchmal vor, daß sich ein Computer aufhängt – da der Bondwell keinen Reset-Taster hat, muß man ihn notgedrungen ausschalten, und schon kann die Arbeit der letzten Stunde verloren sein.

Positiv ist hingegen zu bemerken, daß man das Betriebssystem CP/M 2.2 auch von der RAM-Floppy aus booten kann – dem Einsatz von Disketten ohne System in den ersten Spuren steht also nichts im Wege.

Kontrastarm

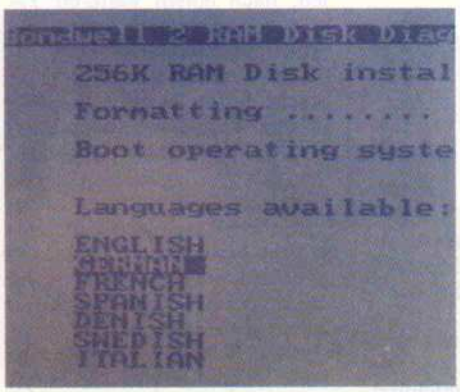
Als problematisch hat sich das LC-Display erwiesen: um eine einigermaßen kontrastreiche Darstellung der Zeichen zu erreichen, muß man den Bildschirm direkt beleuchten. Bedingt durch die starke Beleuchtung kommt es wiederum zu starken Reflexen auf der Schutzscheibe des Displays, so daß es nahezu unmöglich ist, einen Betrachtungswinkel zu finden, der es gestattet, das Display einwandfrei ablesen zu können. Mit dem Regler zur Einstellung des Kontrasts kann man auch keine merkliche Verbesserung

der Lesbarkeit erreichen. Bei sehr ungünstigem Betrachtungswinkel kann man noch durchlaufende Unschärfen erkennen. Alles in allem konnte der Bildschirm nicht überzeugen – er ist der wesentliche Schwachpunkt des Rechners.

Tastatur

Die Tastatur hingegen überraschte positiv: sowohl die Anordnung als auch das 'Feeling' waren gut. Die Anordnung entspricht einer deutschen Tastatur; das 'Z' ist also da, wo es hingehört. Durch die entsprechenden Routinen im Monitor-ROM kann die Tastatur an verschiedene länderspezifische Belegungen angepaßt werden.

Schwierigkeiten bereiteten dagegen die Tastaturschablonen für Standardprogramme. Sieben Pappstreifen soll man in einen oberhalb der Funktionstasten angebrachten engen Schlitz pressen. Sollte man das tatsächlich schaffen, ohne daß mindestens ein Streifen verknickt, so sind damit noch nicht alle Probleme gelöst: zum Ablesen der Schablone müßte man eine Lupe verwenden.



Es ist nicht leicht, das IC-Display so zu beleuchten, daß man die Schrift gut lesen kann.

Verbindungen

An der Rückwand des Bondwell Model 2 befindet sich eine kleine Plastikklappe, hinter der drei Min-D-Steckverbindungen angebracht sind: für eine serielle und eine parallele Schnittstelle sowie für ein zusätzliches Diskettenlaufwerk. Leider entspricht keine der Buchsen einem Standard – man ist also gezwungen, die entsprechenden Kabel von Bondwell zu beziehen, wenn man nicht die Leitungen selbst löten möchte. Den Ergänzungssatz mit den entsprechenden Kabeln aber kann man nur zusammen mit dem Gerät kaufen – so steht es jedenfalls in der Preisliste.

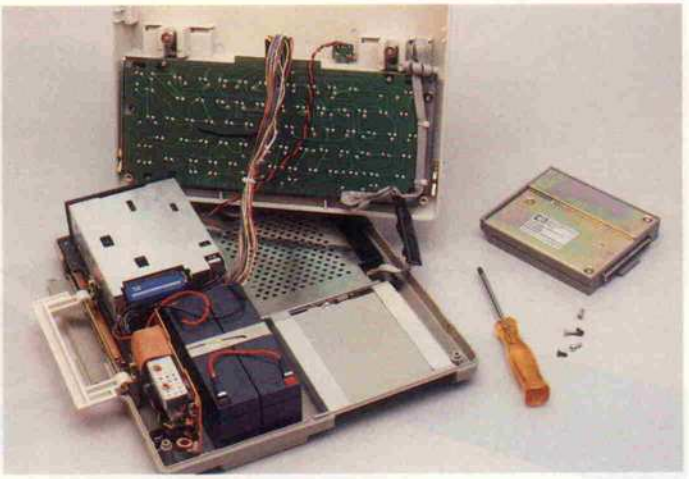
Beim aufmerksamen Studium der Preisliste fällt noch ein anderer Punkt auf: die Handbücher zu den Micro-Pro-Programmen sind für 135 DM als Ringbucheinlagen erhältlich. Da diese aber wohl mit amerikanischer Lochung versehen sein dürften, muß man für weitere 85,00 DM noch die Ringordner 'Original Micro-Pro' dazukaufen.

Grafik ?

Obwohl der Bondwell laut Werbeaussagen auch eine Grafik von 640 × 200 Pixeln bieten soll, konnten wir dem Testgerät keine Grafiken entlocken. Es werden zwar im Benutzerhandbuch die Adressen der eingebauten Peripherie-ICs vermerkt, aber welche Befehle zum Beispiel der LCD-Controller verarbeiten kann, wird nirgends beschrieben. Da man für weitere 145 DM die Kopie des 'Technical-Service-Manuals' kaufen kann, bleibt zu hoffen, daß hier die notwendigen Informationen zu finden sind.

Software

Zu den Micro-Pro-Programmen selbst braucht man wohl nichts mehr sagen – die etwas betagte Software ist ausgereift und dürfte den meisten Ansprüchen genügen. WordStar bleibt das Textverarbeitungsprogramm unter CP/M, für die Programme CalcStar, DataStar und ReportStar gibt es sicherlich Pendants, die besser und schneller sind. Aber durch die Festlegung auf die Programme eines Software-Hauses erreicht man hier ein hohes Maß an Portabilität zwischen den einzelnen Programmen.



Den meisten Platz im Bondwell 2 belegen das Floppy-Laufwerk und die beiden Akkus.

Das Finanzbuchhaltungs-Programm ist jedoch sehr zweifelhaft. Es stürzte jedesmal beim Versuch ab, Buchungssätze neu anzulegen. Dabei zeigte es sich, daß das Programm in Pascal geschrieben wurde: es meldet einen Fehler '0E' bei 'PC 4976H'.

Hardware

Nachdem man vier Schrauben an der Geräteunterseite und zwei Schrauben hinter dem Tragegriff gelöst hat, kann man das Gehäuseoberteil abheben.

Die Computer-Platine ist von einem Metallkäfig umgeben – wohl um die Störstrahlung zu reduzieren. Unter dem Käfig erkennt man, daß Bondwell überall, wo es möglich ist, CMOS-ICs verwendet hat. Für die verbleibenden TTL-Chips sind durchweg HCT-Typen eingesetzt.

Fragwürdig ist hingegen, warum in die Leitung zum Display eine kleine zusätzliche Platine eingefügt ist. Ob hier nachträglich Verbesserungen eingebaut wurden? Auffällig ist die lose im Gehäuse angebrachte Platine für die Spannungswandlung. Wohl um Kurzschlüssen vorzubeugen, war sie mit einem Pappstreifen umgeben.

Die Computer-Akkus sind nicht gegen Überladen geschützt – man muß also nach zwölf Stunden unbedingt den Stecker ziehen.

Resümee

Das Hauptproblem bei dem Bondwell 2 ist eindeutig das mangelhafte Display. Außerdem stören verschiedene Kleinigkeiten bei der Arbeit mit dem Model 2. Da entschädigt einen auch nicht die im Lieferumfang eingeschlossene Software. Die Idee, einen Portable mit einem 3,5"-Laufwerk und einer RAM-Floppy auszustatten, ist gut. Erlaubt diese Konfiguration doch den Zugriff auf größere Datenbestände. Sein Gewicht von 6 kg und seine Maße machen den Bondwell Model 2 zu einem durchaus tragbaren Computer. Ob jedoch ein portabler Computer mit der Z80-CPU überhaupt noch Chancen am Markt hat, wird sich zeigen.

Ergebnisse auf einen Blick

- ⊕ Tastatur gut zu bedienen
- ⊕ gutes Preis/Leistungsverhältnis
- ⊕ viel Software im Lieferumfang
- ⊖ schlecht ablesbares Display
- ⊖ RAM-Disk nicht batteriegepuffert
- ⊖ Handbücher in englischer Sprache

c't

Penman Penplot

Der Mausplotter

Eckart Steffens

'Das ist also ein Plotter, aha.' Das 'aha' klingt so ungläubig, wie es gemeint ist. Beim Anblick der etwas übergroßen Zirkelschachtel ist zunächst wohl auch etwas Skepsis angebracht. Doch intensives Ausprobieren belehrt: Das Ding ist tatsächlich ein Plotter, es funktioniert und bringt all das zu Papier, was seine konventionellen Kollegen auch schaffen.

Penplot besteht aus den folgenden Teilen: zunächst ist da einmal die bereits erwähnte Schachtel – sie läßt sich auseinanderziehen und beherbergt im größeren Teil die Plotter-Steuerung, die mit dem abnehmbaren, fahrbaren Zeichenkopf über ein mehradriges Flachbandkabel verbunden ist. Fahrbar bedeutet, daß an der Unterseite des Zeichenkopfs drei Rollen montiert sind – zwei davon sind angetrieben, die dritte ist nur eine Stützrolle. Der Zeichenkopf wird mit drei Standard-Tuscheeinsätzen bestückt, und dann geht er los, genauer: fährt er los. Denn hier bewegt sich der gesamte Plotter wie eine Schildkröte über das Papier und malt. Wem der Vergleich lieber ist, der kann auch sagen: Wie eine umgekehrte Maus. Damit läge man nicht einmal daneben, denn das Ding ist auch bei der Eingabe als Maus zu gebrauchen.

Außerdem sind im Lieferumfang enthalten: ein Netzteil, ein RS-232-Verbindungskabel, eine Zeichenplatte aus schwarzem Acrylglas und eine Diskette mit Software, die unter MS-DOS auf dem IBM PC (oder Kompatiblen) lauffähig ist. Der Test auf einem Commodore PC-10 ergab keine Probleme. Die Diskette ist nicht kopiergeschützt, jedoch sind die Programme nur auf dem Original-Datenträger lauffähig, der durch einen kleinen Laser-Einbrennfleck mit einem charakteristischen 'Fingerabdruck' versehen ist. Allerdings setzt die Verwendung von Penman das Vorhandensein einer Standard-Grafikkarte voraus.

Nach dem Laden des Programms zeigt ein Begrüßungsmenü je 16 Icons für zwei Programme. 'Paint Box' ist ein Zeichen- und Malprogramm, das zur Gestaltung beliebiger grafischer Vorlagen dient. Man kann Linien, Kreise, verschiedene Pinselstärken, Umrandungen und Flächen wählen; und auch das Freihandzeichnen wird unterstützt. Demgegenüber ist 'Business Graphics' ein Paket zur leichten Erstellung von Balken-, Linien- und Tortendiagrammen mit einer frei wählbaren Gestaltung der Grafiken. Achseneinteilungen werden nach Eingabe der gewünschten Unterteilung automatisch erstellt, Überschriften richtig plaziert und ausgemittelt. Eine 'Spreadsheet'-Funktion gestattet es, Datensätze schnell zu editieren, die man von Diskette als 'BGP' (BusinessGraphics Penman)-Files, 'DIF'-Files oder 'WKS'-Files laden kann. Dadurch kann das Programm Daten aus Programmen wie Lotus, Symphony oder Multiplan laden und mit Penplot ausgeben. Die Darstellung der Grafik entspricht exakt dem Bild, das auch auf dem Schirm sichtbar ist; das heißt, daß auch alle Beschriftungen im richtigen Maßstab eingetragen werden.

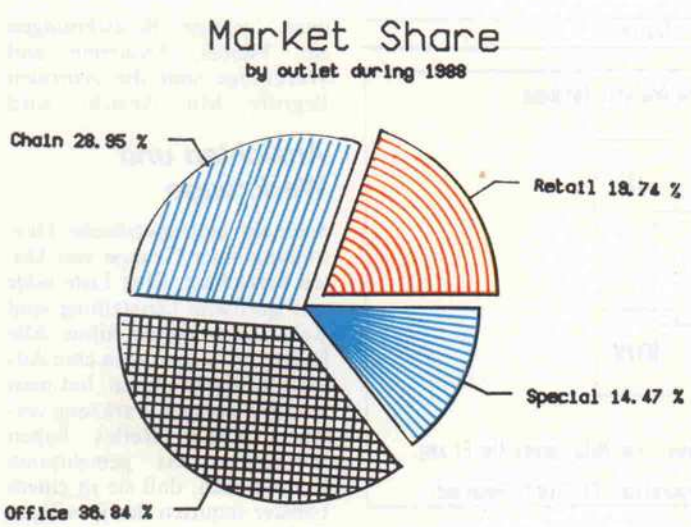
Die Ausgabe kann über einen Epson-Matrixdrucker mit dem Graftrax-Plus Grafikzusatz, einen HP-Plotter 7470A oder aber eben über den Penman Penplot erfolgen. Wählt man Penplot an, erfolgt zunächst die Aufforderung, den Plotter einzurichten; nach einem weiteren Tastendruck beginnt dann der Ausdruck. Nebenbei bemerkt: das Begleit-Manual läßt noch etliche Wünsche offen; so wird in ihm kein Wort (oder Satz) über diese Einrichtprozedur verloren – glücklicherweise kann man sich darüber Informationen über einen Hilfsbildschirm beschaffen. Danach ist es erforderlich, den Penplot auf ein weißes Stück Zeichenpapier zu stellen, das sich auf einer ebenen, schwarzen Unterlage befindet (dafür also die Acryl-Platte).

Sodann positioniert man den Plotter auf dem Papier nahe der oberen linken Ecke. Was nun passiert, ist faszinierend und wiederholt sich während des gesamten Plottvorganges mehrere Male: Der Plotter dreht sich und fährt über die linke Papierkante. Hat er diese durch den Weiß/Schwarz-Übergang identifiziert, stellt er sich exakt parallel zu dieser Linie. Dieser Vorgang wiederholt sich an der oberen Papierkante. Erst dann fährt der Plotter mitten auf die Zeichenfläche und beginnt mit dem Plottvorgang. Es ist also gleichgültig, wie das Papier liegt; die ausgegebene Grafik wird immer gerade auf dem Blatt erscheinen.

Der Bildschirm wird auf DIN-A4-Format ausgeplottet; ein Farbwechsel (drei Stifte führt der Plotterkopf mit) ist durch

entsprechende Eingaben auf Zwischenabfragen hin möglich. Erfolgt keine Eingabe, geht es nach 30 Sekunden mit der alten Farbe weiter.

Durch das ständige Nachzentrieren an den beiden Papierkanten braucht Penplot zur Fertigstellung eines Diagramms erheblich länger als ein konventioneller Plotter. Penplot 'überfährt' die Zeichnung relativ oft, was insbesondere bei langsam trocknender Tusche zum Verschmieren der Zeichnung führen muß. Zu dünnes Papier wellt sich allein schon durch das Eigengewicht von Penplot relativ stark und verringert dadurch die Wiederholgenauigkeit. Sowohl an Tuschepatronen als auch an das Zeichenpapier sind daher hohe Anforderungen zu stellen. Ein entgegen den Empfehlungen des Herstellers durchgeführter Test mit Fotokopierpapier erbrachte für die Balkengrafik akzeptable, für das Tortendiagramm hingegen weniger gute Ergebnisse. Hinzu kommt, daß das Verbindungs-Flachbandkabel zwischen der Steuereinheit und dem Zeichenkopf



Manchmal verschmiert Penman seine eigenen Bilder: Er 'fährt' über die noch feuchte Tinte oder verwischt sie mit seinem Kabel.

über das Papier schleift und das Seinige dazu tut, eventuell noch nicht verwischte Zeichnungsteile zu verschmieren. Möglicherweise wäre hier ein Kabelabgang nach oben eine konstruktiv geschicktere Lösung gewesen.

Wie bereits erwähnt, kann man Penplot auch als Maus benutzen, was die Software bei der Auswahl in den Icon-Menüs auch unterstützt: ein Verschieben des Zeichenkopfes bewirkt hier eine Cursor-Bewegung (er springt in kleinen Schritten) über den Schirm, die Symbole werden sodann mit der Return-Taste 'angeklickt'. Beim Plotten funktioniert die 'Maus' dann wieder als Zeichengerät.

Der Aktionsradius von Penplot wird durch das Verbindungskabel zur Steuereinheit bestimmt und beträgt, eine 3/4-Drehung um sich selbst einbezogen, gut 45 cm. Ein DIN-A3-Format ließe sich also noch ausplotten; die Ausgabe vom Bildschirm nutzt jedoch nur das Format DIN A4 aus. Wie man jedoch einen größeren Bereich nutzt (und gegebenenfalls noch erweitert) steht ebensowenig im Handbuch wie Details über die RS-232-Schnittstellenprogrammierung, die Nutzung aus BASIC sowie die Übertragung von HP-GL auf ein Penplot-Format. Und das ist schade; denn auch wenn Penplot weder zum Erstellen von Präzisionszeichnungen noch zur Beschriftung von Pappkartons unbedingt prädestiniert ist, kann man mit diesem kleinen Kobold sicher mehr machen, als ihn spektakulär im Büro über den Fußboden flitzen zu lassen. ct

T. S. Datensysteme-Vertriebsgesellschaft mbH

Brandneu
für den Spectrum 48 K

Fairlight	29,90
Lord of the Rings	59,90
Yie ar Kung Fu	29,90
Surf Champ (incl. Surfbrett)	49,90
Rasputin	29,90
Spitfire 40	39,90
Robin of the wood	39,90
Winter Sports	34,90
Roller Coaster	26,90
WHAM (Musiksynthesizer)	34,90
A View to a Kill (James Bond)	25,90
Arc of Yesod (SUPER GRAPHIC)	29,90
Sweevo's World	29,90

Spectrum +
solange Vorrat **333,—**

Competition Pro 5000
Der Super-Joystick **49,90**

MULTIFACE ONE
Endlich ist das Problem der Sicherungskopien endgültig gelöst.
Mit Multiface können Sie **jedes** Programm, das Sie in Ihren Spectrum geladen haben, auf Band, Microdrive-Cartridge oder Floppydisk abspeichern. Daneben ist eingebaut: Joystick-Interface, Composite Video-Interface, 8KB ROM, 8KB RAM, den Sie auch sonst nutzen können.
Multiface speichert den gesamten Rechnerinhalt (inkl. Register) auf jedes am Spectrum anschließbare Medium.
Natürlich haben sie 6 Monate Garantie
DM 159,90

BIS ZU 1,4 MBYTE FÜR JEDEN SPECTRUM
Die neuen Discovery-Systeme mit — Centronics-Drucker-Interface — Joystick-Interface (Kempston-Typ) — Video-Monitor Ausgang — stabilisiertem Netzteil (versorgt auch Ihren Spectrum) — Befehlssatz voll Microdrive kompatibel — belegt **keinen** RAM im Spectrum — deutsches Handbuch und Einführungsdiskette — 1/2 Jahr volle Garantie.

DISCOVERY 180
ein Laufwerk
180K Speicherplatz
DM 599,—

DISCOVERY 360
zwei Laufwerke
360K Speicherplatz
DM 848,—

DISCOVERY 720
ein Laufwerk
720K Speicherplatz
DM 899,—

DISCOVERY 1400
zwei Laufwerke
1,4M Speicherplatz
DM 1348,—

DISCOVERY PLUS 18
Einbausatz mit
1 Zusatzlaufw. 180K
DM 289,—

DISCOVERY PLUS 72
Einbausatz mit
1 Zusatzlaufw. 720K
DM 499,—

C-64 NEU!!!!

Rock'n Wrestle (Cass. Original Spiel)	39,90
Rock'n Wrestle (Disk Original Spiel)	49,90
Back to the Future (Cassette)	39,90
Rambo (Cassette Original Spiel)	34,90
Commando (Cassette Original Spiel)	39,90
Mercenary (Cassette Original Spiel)	39,90
Koronis Rift (Cassette Original Spiel)	39,90
Koronis Rift (Disk Original Spiel)	49,90

Sinclair QL

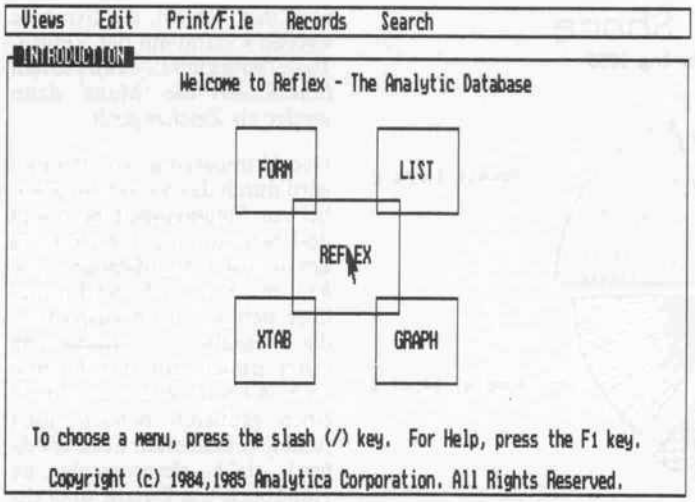
Extended Basic (ROM-Modul)	99,90
QL-Monitor (ROM-Modul)	99,90
QL-Hyperdrive (3-D-Autorennen)	59,90
QL-Match-Point (3-D-Tennis)	59,90
QL-Chess (Schachprogramm neue Version)	79,90
QL-West (Adventure)	69,90
QL-Lost Kingdom of Skull	69,90
Flugtrainer (deutsch)	49,90
Cartridge Doctor	59,90
Snooker (Super Billard)	59,90
Kempston Diskinterface (mit Standard-Shugart-Bus) komplett einsteckbar	399,90
Centronics Drucker Interface	129,90

Cartridges Disketten

Microdrive Cartridge (Sinclair)	8,90
Diskette 3½" je 10 Stück	89,00
Reinigungsdisk-Set 3½"	59,90
Diskette 5¼" je 10 Stück	27,90
Disketten Box 5¼" für 70 Stück	35,90
Disketten Box 3½" für 40 Stück	35,90

Alle Preise enthalten die gesetzliche Mehrwertsteuer. Versand per Nachnahme (zuzüglich 5,90) oder Vorkasse mit Scheck (zuzüglich 2,50).
Informationen können angefordert werden. Händleranfragen erwünscht.

Denisstraße 45, 8500 Nürnberg 80, Tel. 09 11/28 82 86



Neu von Borland: 'Reflex'

Datenanalyse

Peter Glasmacher

Da hat man nun eine relationale Datenbank, eine Tabellenkalkulation und noch einiges an guter Software, um der progressiv anwachsenden Datenmenge Herr zu werden. Nach einiger Zeit hat man nun seine Daten im Griff, stellt aber mit Entsetzen fest, daß sich die gerade für eine Analyse benötigten Daten nicht in der Tabellenkalkulation, sondern in der Datenbankdatei befinden. Natürlich sind die Dateiformate völlig inkompatibel, und es besteht fast keine Chance, beide Dateien miteinander zu vergleichen.

In diesen Momenten wünscht man sich ein Programm, das tolerant gegenüber den diversen Formaten ist und zumindest problemlos gängige Dateiformate verarbeitet.

Es scheint, als würde mittlerweile ein derartiges Programm auf dem Markt sein: Die Rede ist von Reflex, einem der Programmpakete, die die Firma Borland jüngst auf den Markt gebracht hat. Reflex sieht auf den ersten Blick wie eines der vielen mehr oder weniger guten Programme zur Tabellenkalkulation aus. Auf den zweiten Blick merkt man jedoch sehr schnell, daß Reflex mehr ist.

Ein rundes Paket

Reflex erreicht den Anwender als Paket, das aus drei Disketten und einem voluminösen Handbuch besteht. Die Disketten enthalten neben Reflex eine Reihe von Beispielprogrammen. Diese Software begleitet den potentiellen Reflex-Profi während des gesamten ersten Teils des Handbuchs und verdeutlicht ihm die Arbeitsweise des Programms.

Doch nun zu Reflex. Nachdem man die Installation durchgeführt hat, überrascht Reflex mit einem Eingangsmenü, das sehr stark an GEM-Programme erinnert. Eine Menüzeile offeriert eine Reihe von Einträgen – und wer eine Maus sein eigen nennt (und den Treiber geladen hat) bemerkt den typischen Pfeil unübersehbar in der Mitte des Bildes. Nun brauchen diejenigen, die dieses 'Tierchen' nicht ihr eigen nennen, nicht zu verzweifeln: die von Programmen wie Lotus 1-2-3 bekannte Tastenkombination 'Schrägstrich' mit den Cursor-Tasten sorgt ebenfalls für einen Zugriff auf die Menüs.

In der Datenverarbeitung spricht man üblicherweise von Bildschirmmasken und Hilfsprogrammen. Reflex führt hier neue, 'griffige' Bezeichnungen ein: Menüs, Ansichten und Werkzeuge sind die zentralen Begriffe. Mit 'Ansicht' wird

Ansichten und Werkzeuge

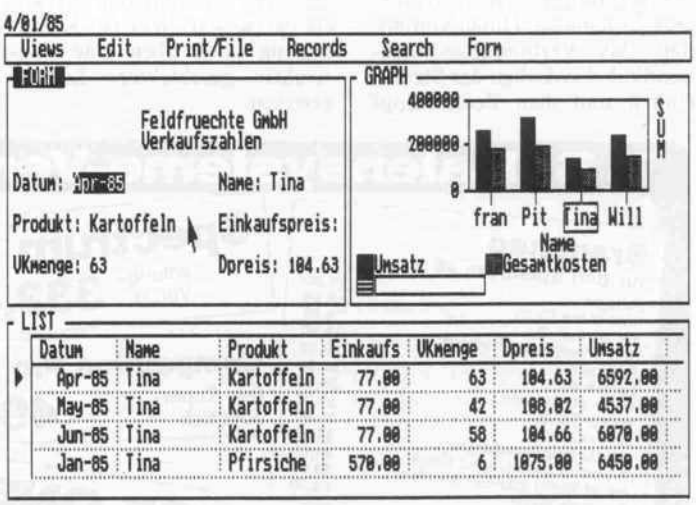
nunmehr eine spezifische Darstellung einer Gruppe von Daten bezeichnet. Eine Liste oder eine grafische Darstellung sind Ansichten in diesem Sinne. Alle Dinge, mit denen man eine Ansicht bearbeiten kann, hat man mit dem Namen Werkzeug versehen. Unter Reflex haben Werkzeuge das gemeinsame Kennzeichen, daß sie in einem Fenster inmitten des jeweiligen Bildes auftauchen und interaktiv die benötigten Einstellungen abfragen.

Die Stärke von Reflex ist seine Fähigkeit, Daten schnell und übersichtlich unter verschiedenen Gesichtspunkten zu analysieren. Bei der Ausgabe der Ergebnisse kann der Anwender zwischen einer Liste, einer grafischen Darstellung und einem Formular wählen. Diese drei Ansichten lassen sich zudem gleichzeitig in drei Fenstern betrachten. Mehr noch, die Veränderung eines Wertes in einem beliebigen Fenster sorgt für eine sofortige Aktualisierung aller anderen Ansichten.

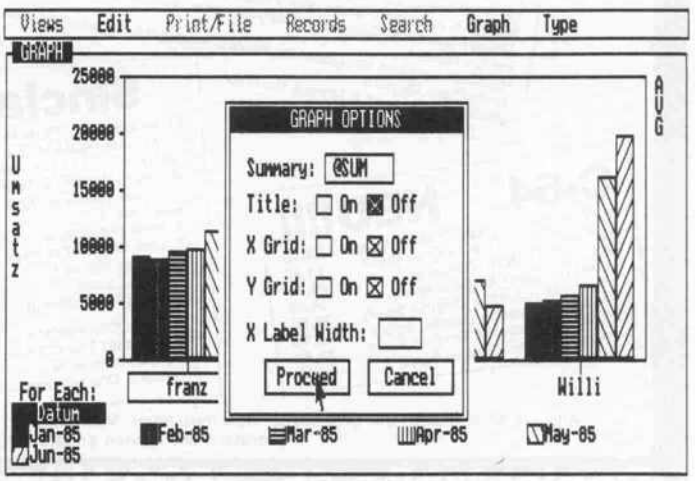
Analysen

Berechnete Felder sind ähnlich wie in Tabellenkalkulationen erlaubt. Eine umfangreiche Sammlung von eingebauten Funktionen, vorwiegend aus dem finanztechnischen und kaufmännischen Bereich, erleichtert den Aufbau komplexer Formeln.

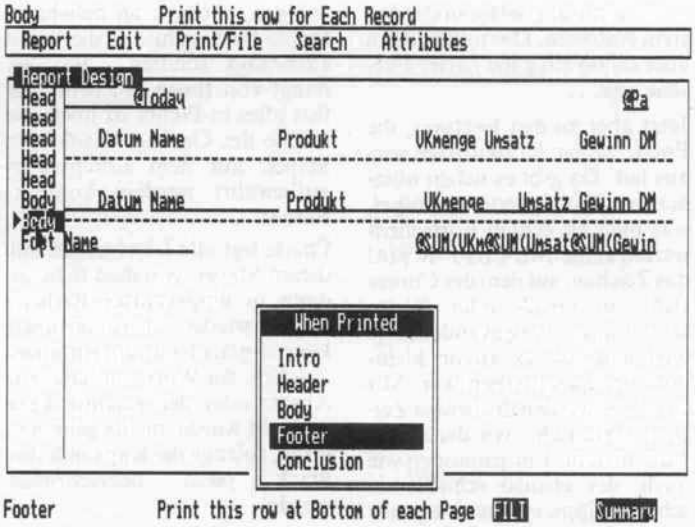
Weiterhin kann man Datenbestände nach verschiedenen Gesichtspunkten sortieren, filtern



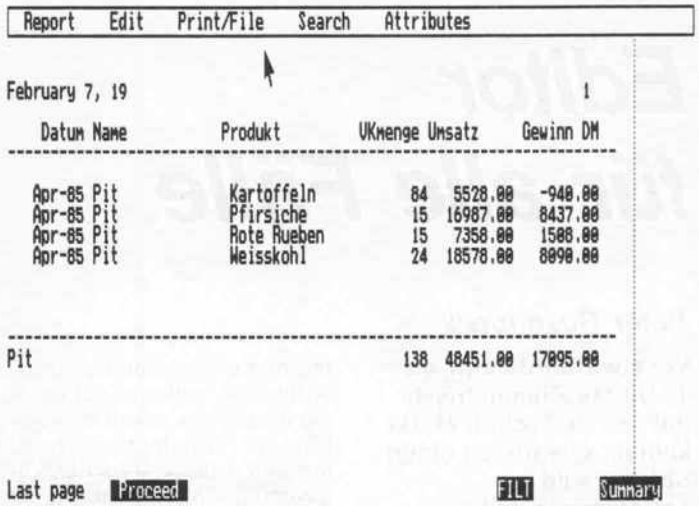
Drei Reflex-Ansichten: Formular, Liste und Grafik auf einen Blick



Die Grafik-Ansicht mit einem 'Werkzeug'



Der Listengenerator und . . .



. . . die dazugehörige Liste

und aus dem Bestand die gewünschten Informationen als Extrakt herausziehen. Alle diese Tätigkeiten werden mit eigenen Werkzeugen durchgeführt (man kann sich an die Terminologie leicht gewöhnen).

Wer keine Maus hat (oder sie nicht benutzen möchte) kann Reflex auch über die zehn Funktionstasten steuern, die als Zierde eines Rechners mal auf der linken Seite, mal oberhalb des Tastenfeldes angeordnet sind. Die Belegung der Tasten verrät dem Anwender, daß sich die Software-Entwickler dabei durchaus etwas gedacht haben. Oft benötigte Funktionen kann man bequem erreichen, Funktionen mit einschneidenden Auswirkungen liegen entsprechend ungünstiger. So sorgt die F10-Taste dafür, daß ein Werkzeug geöffnet wird. Dieses Werkzeugt zeigt nun eine Auswahl der gerade aktuellen Feldnamen, die man einfach mit Hilfe der Cursor-Tasten auswählen kann. Leider zeigt sich auch hier die Inkompatibilität der landesspezifischen Tastaturen. PC-Besitzer im angelsächsischen Raum erreichen mit dem kleinen Finger den Schrägstrich, jene bedeutungsvolle Taste, die erst den mauslosen Zugriff zu den Menüs ermöglicht. Der deutsche Anwender hat jedoch nur über die Shift-Taste und einer (visuellen) Suchoperation auf der obersten Tastenreihe Zugriff auf den so wichtigen Schrägstrich.

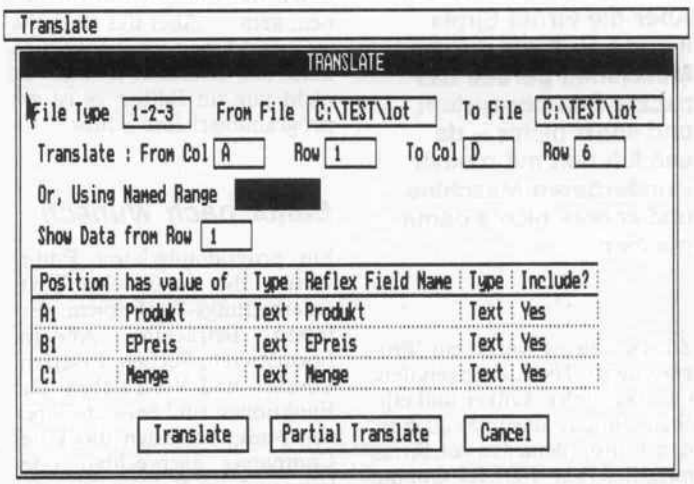
Gelistet

Nun sind die Verkaufsergebnisse oder die Bilanz auf dem Bildschirm in verschiedenen Ansichten zwar recht schön, aber wer schleppt seine Hardware mit zu einer Sitzung oder zum Chef-Rapport. Dafür bietet Reflex einen komfortablen Reportgenerator, der das Erstellen und Drucken beliebiger Listen ermöglicht. Der Listenproduzierer befindet sich in einem Programm des Reflex-Paketes, das unabhängig von Reflex geladen wird. Dieser Listengenerator ermöglicht es, sich das Ergebnis bereits nach dem 'Design' der Liste auf dem Bildschirm anzuschauen – eine Einrichtung, die viel Zeit und noch mehr Papier spart.

Zu Listen aller Art gehören auch entsprechende Grafiken – also ist ebenfalls ein Grafik-Druckprogramm in Reflex integriert. Auch eine fertiggestellte Grafik kann man vor dem Druck auf dem Bildschirm betrachten. Diese Funktion wird wohl jeder schätzen, der seinen Drucker im Punktmodus quälen muß.

Toleranz

Reflex ist ein neues Programm, und seine Entwickler hätten sich nun sagen können, daß es auf ein Dateiformat mehr oder weniger nicht ankommt. Die Reflex-Entwickler sind jedoch einen anderen Weg gegangen: Reflex ist in der Lage, normale ASCII-Dateien und Formate wie dBase, DIF und Lotus 1-2-3 Dateien zu lesen und den Reflex-Ansichten verfügbar zu machen. Der geplagte Analysierer muß somit seinen Datenwust nicht mehr mühselig neu eintippen.



So wird eine Lotus-Datei übersetzt.

Dokumentation satt

Zu Reflex gehört ein Handbuch, das die Bezeichnung Buch zu Recht trägt: in acht Kapiteln auf über 400 Seiten wird der Anwender Schritt für Schritt in die Geheimnisse von Reflex eingeführt. Ein gut gemachtes, mit vielen Beispielen versehenes Tutorial vermittelt anschaulich, wie man mit den Reflex-Ansichten und Werkzeugen ökonomisch umgeht. Das vorliegende Werk ist zwar in englischer Sprache verfaßt, laut Auskunft des Anbieters soll jedoch in Kürze auch eine deutschsprachige Version erhältlich sein.

Zu guter Letzt

Reflex zeigt neue Wege, Daten zu analysieren und Relationen anschaulich darzustellen. Die gesamte Bedienung des Programms läßt erahnen, daß Drop-Down-Menüs und Mäuse in Zukunft unentbehrliche Werkzeuge eines Computer-Anwenders sein werden. Reflex ist jedoch kein relationales Datenbanksystem wie dBase und bietet auch keine Tabellenkalkulation wie Lotus. Trotzdem werden die Datenbank und die Kalkulation den meisten Anwendungsfällen gerecht. Die Stärke von Reflex liegt vielmehr in der Datenanalyse.

Hervorzuheben ist außerdem der recht geringe Preis von Reflex: er liegt mit rund 450 DM nur bei ungefähr einem Viertel des Preises vergleichbarer Programme. Reflex ist also auch hinsichtlich seines Preises schon ein starkes Stück.

Bezugsquelle:
Heimsoeth, Fraunhoferstr. 13, 8000 München 5.

Pmate

Editor für alle Fälle

Peter Rosenbeck

Vor etwa fünf Jahren, als 16-Bit-Maschinen frisch auf den deutschen Markt kamen, erwarb ich einen Sirius – wild entschlossen, tolle Programme für diese Maschine zu entwickeln. Aber die Firma Sirius lieferte zu ihren Maschinen gerade das nackte Betriebssystem und sonst nichts – da saß ich nun mit meiner wunderbaren Maschine und konnte nichts damit machen.

Zu dem Sirius wurde ein 'Programmer's Toolkit' angeboten, mit Assembler, Linker und etlichen Utilities, darunter auch einem Editor (denn den von Sirius mitgelieferten EDLIN konnte man ja nur als Zumutung bezeichnen). Biß ich also in den sauren Apfel und blätterte nochmal ein paar Hunderter auf den Tisch.

Warum ich Ihnen das erzähle? Weil ich dadurch meinen ersten Kontakt mit Pmate hatte. Zuerst arbeitete ich mit WordStar als Editor. Aber da ärgerte mich schon bald der für eine 16-Bit-Maschine lächerlich langsame Bildaufbau. Aus purer Neugier kramte ich deshalb mein Pmate-Exemplar von den Disketten des Sirius-Entwicklungssystems herunter – und seitdem ist Pmate mein Leib- und Mageneditor.

Der blitzschnelle Bildaufbau war der Auslöser (Pmate schreibt direkt in den Bildschirmspeicher): bei einem 40 bis 50 KByte großen File kommt man in einem Augenblick vom Anfang ans Ende – denn Pmate hat einen ernstzunehmenden Textpuffer. Und da nahm ich mir eben mal das Handbuch vor.

Davon war ich sofort fasziniert: Nicht etwa, weil es so gut gewesen wäre. Nein, es war so ziemlich das Härteste, was mir jemals auf dem Gebiet der Dokumentation untergekommen ist. Da war kein Satz zuviel, kein überflüssiges Beispiel unterbrach die Flut an Informationen, kein... Aber das stachelte meinen Ehrgeiz an. Schließlich kam ich dahinter: Pmate ist nicht nur ein Editor, es ist ein programmierbarer Editor.

Editor nach Wunsch

Ein programmierbarer Editor verfügt über die üblichen Textverarbeitungs-Funktionen: Erfassen, Betrachten, Ändern, Formatieren, Suchen, Ersetzen, Löschen und Verschieben. Die Funktionen sind entweder über den Funktionstasten-Block des Computers ansprechbar oder können durch Kombination der Control- mit Buchstaben-Tasten ausgelöst werden. Das kennt man von anderen Systemen, allen voran von WordStar. Allerdings: Pmate ist mit WordStar nicht kommandokompatibel. Jedenfalls nicht so, wie er ausgeliefert wird. Aber Pmate ist konfigurierbar, und das in einem Ausmaß, daß Oldies wie WordStar so alt aussehen, wie sie auch sind.

Es gibt in Pmate – mit einer wesentlichen Ausnahme – alles, was WordStar zu bieten hat und dazu einiges mehr. Die Ausnahme betrifft die Textformatierung: Pmate ist ein Editor und kein Textsystem. Das Programm macht keinen Blocksatz, es unterbreitet keine Trennungsvorschläge und unterstützt direkt keine Features eines Druckers. Die fehlende Drucker-Unterstützung ist aufgrund der Programmierbarkeit von Pmate kein prinzipielles Problem. Rein theoretisch könnte ein versierter Pmate-Benutzer auch Blocksatz und Silbentrennung selbst in das System einbauen. Das müßte dann aber schon ein ganz harter Bursche sein...

Jetzt aber zu den Features, die Pmate seiner Konkurrenz voraus hat. Da gibt es neben nützlichen Kleinigkeiten einiges, was man als echten Fortschritt werten kann. Mit Ctrl-F-R wird das Zeichen, auf dem der Cursor steht, in Groß- oder Kleinschreibung umgewandelt, je nachdem, ob es zuvor klein- oder großgeschrieben war. Mit Ctrl-F-C werden die beiden Zeichen vertauscht, vor denen der Cursor steht. Für jemanden wie mich, der ebenso schnell wie schlecht tippt, ist das eine große Erleichterung. Mit Ctrl-S wird jedes nachfolgende Kommando wiederholt, wobei die Anzahl der Ctrl-S den Wiederholungsfaktor bestimmt: einmal Ctrl-S wiederholt viermal, zweimal wiederholt 16mal, dreimal wiederholt 64mal... sehen Sie den Dreh? Andere Wiederholungsfaktoren sind auch kein Problem; man muß dafür jedoch die Kommandosprache bemühen.

Elf Texte gleichzeitig

Pmate hat jedoch noch mehr zu bieten: das System verwaltet bis zu elf Textpuffer zugleich. Das bedeutet, daß man gleichzeitig elf Dateien bearbeiten kann. Allerdings wird nur für einen Puffer (den Textpuffer oder Puffer 'T'; die andern Puffer sind von 0 bis 9 durchnumeriert) die Diskettenpufferung automatisch vorgenommen. Das bedeutet: ist eine Datei größer als der für ihren Puffer verfügbare Speicherplatz, lädt Pmate nur soviel von der Datei in den Speicher, wie Platz ist. Möchte man jetzt aber Text bearbeiten, der gerade nicht im Speicher steht, dann sorgt Pmate im Puffer 'T' automatisch dafür, daß ein nichtbenutzter Teil der Datei auf Diskette geschrieben und der Rest nachgeladen wird. Bei anderen Puffern muß man dies von Hand besorgen (aber dafür gibt's Kommandos).

Beim WordStar löscht man Zeilen mit Ctrl-Y – eine Tastenkombination, die man auf DIN-Tastaturen manchmal ungewollt betätigt. Da hilft dann nur mehr Neutippen.

Nicht so in Pmate. Der merkt sich alle Löschungen und kann sie auf Tastendruck (Ctrl-L) wieder in den Text einfügen. Und nicht nur da, wo sie zuvor standen, sondern an beliebiger Stelle. Und nicht nur die letzte Löschung, sondern – tja, das hängt von Ihnen ab, denn wie fast alles in Pmate ist auch die Größe des Garbage-Stack (der Stapel, auf dem Löschungen aufbewahrt werden) konfigurierbar.

Pmate legt alle Löschungen auf diesen Stapel, von dem man sie dann in umgekehrter Reihenfolge wieder herunterholen kann. Egal, ob nun ein einzelnes Zeichen, ein Wort, ein Satz, ein Absatz oder der gesamte Text gelöscht wurde: nichts geht verloren, solange die Kapazität des Stack nicht überschritten wurde.

Zeichen, Worte und Zeilen sind Einheiten, die der Editor für die Cursor-Bewegung und für das Löschen unmittelbar 'versteht'. Für die Verarbeitung größerer Einheiten muß man diese als Blöcke markieren. Blöcke kann man löschen, verschieben, aber auch in andere Puffer bewegen. Überhaupt ist der Austausch von Informationen zwischen verschiedenen Puffern unproblematisch. Nur eines würde sich der verwöhnte Rezensent im Zusammenhang mit der Puffer-Verarbeitung wünschen: Windowing. Denn ohne sieht man immer nur einen Puffer auf dem Bildschirm.

Man kann eine Marke im Text setzen und den Cursor blitzschnell zu dieser hinbewegen. Dabei stört es mich ein wenig, daß in der Grundausstattung nur eine Marke gesetzt werden kann. 'In der Grundausstattung' deshalb, weil nichts den Eingeweihten hindert, sich Pmate so zu dressieren, daß er sich auch mehrere Marken merkt.

Pmates Sprache

Womit ich bei der Programmiersprache von Pmate wäre; der eigentlichen Attraktion.

Bei der normalen Textverarbeitung werden Editierfunktionen durch Control-Tastenkombinationen ausgelöst; in der Pmate-Terminologie heißen diese 'unmittelbare' Kommandos. Jedes unmittelbare Kommando steht jedoch auch noch in anderer Form zur Verfügung. Das ist die Form, die für den Kommando-Modus gedacht ist.

Mehrere dieser Kommandos können zu 'Makros' zusammengefaßt werden. Die Makro-

```
^xc  ;  kommentiere eine Zeile (Makro erhaelt den Namen "c")
1 -m ;  gehe an das Ende der Zeile
@x>50[(@x+2)v1][50v1] ; bewege Cursor entweder auf Spalte 50 oder zwei
0qy                   ; Spalten hinter das Zeilenende
@1QX
i/*$ ;   fuege Kommentar-Anfang ein
77qx ;   gehe zu Spalte 77
i*/$ ;   fuege Kommentar-Ende ein
1    ;   gehe zur naechsten Zeile
;   ENDE ENDE ENDE ENDE ENDE

^xk  ;   Kommentiere Programm (Makro erhaelt den Namen "k")
@1v7 ;   Speichere Zeilennummer des Programmendes in Variable 7
q#   ;   gehe zum Programm-Anfang
@1v8 ;   Speichere Zeilennummer in Variable 8
(@7-@1)[.c];  bestimme Anzahl der Programmzeilen und fuehre "c"-Makro
           ;  entsprechend oft aus
(@1-@8)[-1];  gehe zurueck zum Programm-Anfang
s/*$1m     ;  setze Cursor in ersten Kommentar
;   ENDE ENDE ENDE ENDE ENDE
```

Ein Pmate-Makro

Sprache von Pmate zeichnet sich dadurch aus, daß sie die Wiederholung und die bedingte Auswertung von Kommandos, Kommandofolgen und Makros zuläßt. Anders gesagt: es handelt sich um eine eigene Programmiersprache, komplett mit Kontrollstrukturen, Schleifen, Unterprogrammaufrufen und was sonst so alles dazugehört.

Die Makro-Sprache von Pmate läßt keine Wünsche hinsichtlich ihrer Leistungsfähigkeit offen. Mit ihr ist sogar Random-Zugriff auf Dateien möglich sowie der direkte Zugriff auf Speicherstellen (auch bei dem 8086). Dazu setzt man mit einem Kommando das Segment und mit dem anderen das Offset für den Zugriff. Der Programmierer kann Werte in maximal 100 Benutzer-Variablen speichern; und es steht ihm auch ein eigener Stack zur Verfügung.

Den gesamten Textpuffer kann man beliebig manipulieren; ebenso alle internen Parameter von Pmate. Ich habe noch kein System gesehen, das dermaßen offen ist – aber der Programmierer muß die richtige Sprache sprechen.

Und die ist grauenvoll; Kommandos (es gibt mehr als 160) bestehen bestenfalls aus zwei Buchstaben. Auch bei selbstprogrammierten Makros ist man auf einbuchstabige Namen beschränkt.

Schleifenbildung und bedingte Auswahl sind zwar möglich: alle nötigen Sprachelemente sind da. Aber 'for' und 'while' und 'if. . .then. . .else' versteht man wenigstens noch; die Pmate-Schleifen und Bedingungen hingegen: nichts als eckige Klammern.

Dennoch: es lohnt sich, die Sprache zu erlernen. Denn damit lassen sich Tools erstellen, die besonders dem Programmierer viel Arbeit abnehmen können. Dies war auch der Hauptgrund, warum ich trotz der abschreckenden Sprache und der nicht minder bedenklichen Dokumentation bei Pmate geblieben bin.

Komfort mit Macros

Nach stundenlanger Dechiffrier-Arbeit ('Was kann der Autor bloß gemeint haben. . .?') hatte ich Pmate endlich im Griff. Als Belohnung konnte ich bereits zu einer Zeit mit Desktop-Utilities arbeiten, als noch niemand etwas von Sidekick gehört hatte – meine Utilities waren selbstprogrammiert, und zwar in Pmate.

So liefert ein Pmate-Befehl mit dem Namen eines Anzurufenden als Argument die Telefonnummer einschließlich Adresse. Wohlgemerkt: während des Editierens und ohne den Text verlassen zu haben. Auf Knopfdruck verschwindet die Adresse, und der alte Text erscheint wieder. Das bewirkt ein Pmate-Makro.

Die Umwandlung Hexadezimal-Binär-Dezimal-Oktal während des Programmierens ist auch kein Problem: ein ziemlich kleines Pmate-Makro sorgt für die Wandlung.

Wie weit man mit den Makros kommen kann, zeigen drei Pmate-Varianten, die zum Lieferumfang gehören. Die eine ist eine Menü-Version, die im wesentlichen das Handbuch als On-Line-Dokumentation per Menü abfragbar macht. Die beiden anderen sind Sprachen-Editoren: sie verstehen Fortran beziehungsweise C.

Spezialversionen

Erstmal zu Mmate, der Menü-Version. Sie kann den Benutzer mit den Mängeln der Dokumentation teilweise wieder aussöhnen, denn sie ist wirklich ein gutes Mittel, um die Editierfunktionen von Pmate zu erlernen. Wichtig ist hier vor allem, daß man sich bei der Arbeit nicht die teilweise obskuren Abkürzungen der Pmate-Kommandos merken muß. Alles ist über Funktionstasten und Menü zugänglich (auch die Makro-Programmierung).
Aber: die Hilfestellung wird um so weniger, je mehr man sich in Bereiche der Makro-Programmierung wagt.

Cpmate (die Version, die C versteht), hilft dem Programmierer nicht nur beim Erstellen des Programms: auf Knopfdruck werden C-Schlüsselwörter in den Text eingefügt, komplett mit allen Klammern und hübsch formatiert. Man kann über Parameter sogar eins von drei C-Formaten wählen. Cpmate hilft auch bei der Fehlersuche: Fehlermeldungen des Compilers werden automatisch in das Quellprogramm eingemischt und können dann per Knopfdruck aufgesucht und verbessert werden. Anschließend bereinigt Cpmate die Quelle von den Fehlermeldungen.

Allerdings scheint Cpmate auf den Lattice-Compiler geeicht zu sein; das Format, in dem der Aztec-Compiler seine Meldungen ausgibt, bringt das System durcheinander. Dies kann man natürlich beheben: die Quelle für alle Makros wird mitgeliefert, und man braucht 'nur' die entsprechenden Makros umzuschreiben.

Andere Features jedoch funktionieren unabhängig vom verwendeten Compiler. 'Include'-Dateien, globale Variablen und Funktions-Deklarationen kann man betrachten, Blöcke (im Sinne von C) als Ganzes unter Erhaltung der Formatierung verschieben und kopieren.

Ein Utility muß man als Programmierer ganz einfach haben: XAPMATE und EB. Das sind zwei Programme, mit denen Multitasking möglich ist. Das bedeutet, daß man während eines Compiler-Laufs (im Hintergrund) im Vordergrund mit Pmate an einer anderen Quelle arbeiten kann. Das geht aber anscheinend nur auf sehr engen IBM-Verwandten. Ich habe XAPMATE auf einem (ziemlich kompatiblen) M24 getestet – es lief. Auf einem angeblich voll kompatiblen Osborne 5 verweigerte das Programm den Dienst.

Für IBM-Besitzer mit Maus (zu denen ich nicht gehöre) wird auch noch eine Mausversion mitgeliefert: Cursor-Bewegungen sollen dann mit der Maus ausgeführt werden, und man braucht sich nicht die unmittelbaren Kommandos von Pmate zu merken.

Editor für Kenner

Natürlich gibt es auch an Pmate einige Kritikpunkte: die Dokumentation ist, obwohl umfangreich, viel zu knapp.

Wer bereits einmal mit einem Editor gearbeitet hat, der wird die Kapitel über die unmittelbaren Kommandos (die Ctrl-Sequenzen) wahrscheinlich ohne Probleme verstehen. Aber die Kapitel über die Makro-Programmierung sind Musterbeispiele für Programmierer-Dokumentation. Damit meine ich Dokumentation, die der Entwickler selber verfaßt hat: für ihn selbst ganz praktisch, und

die anderen Leser sollen sich gefälligst anstrengen.

Wer noch nicht programmieren kann, der hat keine Chance, 'Pmate' zu lernen. Wer programmieren kann, muß sich auch erst einmal daran gewöhnen, daß eine Sprache, die Text manipuliert, anders zu nehmen ist als eine universelle Programmiersprache.

Als mir nun die neue Version 4.0 von Pmate in die Hände kam, war ich erstmal guter Dinge. Der Umfang der Dokumentation war im Verhältnis zur alten Version auf beinahe das Dreifache angewachsen. Dann stellte sich aber heraus, daß der Löwenanteil dieses Zuwachses auf das Konto von Anhängen geht, die Spezialversionen von Pmate, den Multitasker und die Maustreiber betreffen. Zu der eigentlichen Pmate-Dokumentation ist nur ein Kapitel hinzugekommen. Dieses erläutert zehn Makro-Beispiele etwas ausführlicher; aber kein Grund zu voreiliger Freude! Alle Beispiele sind in früheren Kapiteln schon mal vorgekommen; lediglich ihre Erläuterung ist in diesem Kapitel nicht so knapp; dafür sind Fehler drin.

Die Unterlagen zur Testversion waren übrigens in Englisch; wenn Sie diesen Artikel lesen, sollte jedoch bereits eine deutsche Version auf dem Markt sein. Vielleicht hat sich der Übersetzer erbarmt und einige zusätzliche Erklärungen hinzugefügt.

Was sich nicht ändern läßt, ist die Makro-Sprache selbst – die Kommandonamen lassen keinen Rückschluß auf ihre Funktion zu. Zwar gibt es eine bestimmte Grund-Systematik: Kommandos, die mit 'B' beginnen, haben mit Puffern zu tun, solche mit 'X' mit Ein-/Ausgabe (jedenfalls die meisten) und die mit 'Q' setzen irgendwelche Parameter. Aber darauf kann man sich nicht immer verlassen.

Schönheitsfehler

Die anderen Kritikpunkte sind weniger bedeutsam, wenn auch manchmal ärgerlich. So ist es zwar möglich, in Pmate die bearbeitete Datei auszudrucken, aber nicht im Hintergrund: Man kann nicht weiterarbeiten, sondern muß warten, bis der Drucker fertig ist. Mit dem erweiterten IBM-Zeichensatz wird Pmate nicht fertig: das scharfe Ess (Character 255) ist ein eigenes Steuerzeichen und kann nicht in den Text eingefügt werden. Bei der Umwandlung von IBM-Zeichen mit Ctrl-F-C macht das System Fehler (aber dafür kann ich mit einem Pmate-Makro zwischen ASCII- und IBM-Zeichensatz hin- und herwandeln).

Schließlich ist Pmate bei der Ausführung von Makros, die den Text manipulieren, nicht gerade schnell. Die Kommentierung von 200 Zeilen dauerte auf dem M24 (Intel 8086, 8 MHz) immerhin elf Sekunden – bei der Geschwindigkeit, mit der der Editor sonst seine Arbeit erledigt, ist das unerklärlich langsam.

Dennoch komme ich zu dem Gesamturteil 'empfehlenswert'. Für Programmierer stellt das System eine große Arbeitserleichterung dar, auch wenn sie nicht in C programmieren. Auch für die Textverarbeitung kann man Pmate einsetzen. Aber für die druckfertige Textaufbereitung greife ich dann doch lieber zu anderen Systemen. Es ist kein Problem, einen mit Pmate erfaßten Text so aufzubereiten, daß man ihn mit 'Word' oder einem anderen System mit annehmbaren Formatier-Fähigkeiten weiterverarbeiten kann.

Ich kann mir jedoch gut vorstellen, daß man Pmate auch für Verwaltungsaufgaben einsetzen kann, die nicht ausschließlich mit Textverarbeitung zu tun haben. Da das System Formulare ausfüllen kann, wären Einsatzgebiete wie Fakturierung und Lagerverwaltung durchaus denkbar.

Gemessen am Preis für WordStar wird der Editor Pmate fast verschenkt: die deutsche Version ist für 580 DM, erhältlich, die englische ist um 50 DM billiger. Pmate ist erhältlich bei:
Comfood GmbH, Flaßkamp 24, 4400 Münster. c't

Der Traum für Manager und Selbständige

VIP Professional™ Power für Ihren Atari ST™ in deutsch

1.Kalkulation
2.Datenbank
3.Grafik

VIP verknüpft Werte aus der Kalkulation mit Eintragungen der Datenbank

VIP generiert z. B. eine Tortengrafik aus Kalkulationsergebnissen

VIP stellt alle Informationen für Ihre Entscheidungen bereit

unverbindl. Preisempf.
DM 748.–

Sie erhalten VIP auch bei Ihrem Händler oder bei uns. Fragen Sie uns auch nach anderen Programmen!!

Folgende Ergänzungs-Programme erwarten Sie in Kürze:

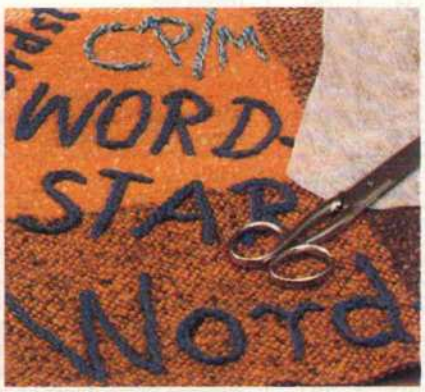
VIP Freelance – eine Textverarbeitung, die effektiv Kalkulations-Werte, Datenbank-Felder und Grafiken übernimmt (z. B. Adressen)

VIP Analysis – ein Statistik- und Analyse-Programm, das die Informationen aus VIP Professional integriert und verarbeitet

CTK Computer Technik Kieckbusch
Am Seeufer 22, 5412 Ransbach, Tel. 02623/1618

Patching WordStar – und kein Ende

Die wichtigsten Patch-Adressen aus vier verschiedenen WordStars



Axel Unterschütz
Detlef Grell

Sie meinen, bei WordStar kennen Sie schon alles? Wußten Sie auch, daß es für CP/M-80 eine Version 3.3 gibt, die – denkbar knapp kommentiert – zum Beispiel beim Dr. Osborne-Kit mitgeliefert wird? Oder kennen Sie sich mit den WordStar-Versionen für CP/M-86 oder PC-DOS aus? Selbstverständlich können wir hier nicht vier Installations-Manuals abdrucken, aber ein bißchen Einblick können wir Ihnen dennoch verschaffen.

Das schicke englische Wort 'patch' heißt 'Flicken'. Vermutlich kennen Sie den Begriff 'Patchwork' als Bezeichnung für eine Technik, in der man Kleidungsstücke oder Decken herstellt. Und das liegt gar nicht so weit ab von dem, was man Programmen beim Patchen antut. Ohne Neukompilierung oder erneute Assemblierung werden Programme verändert, indem einfach bestimmte Speicherzellen verändert werden.

Das ist ein völlig legitimes Tun, und der WordStar beispielsweise ist eigens dafür eingerichtet: er enthält eine beinahe riesige Patch-Area (2 bis 3,5 KByte, je nach Version), in der der forschende Programmierer herumstochern darf. So ziemlich alles, was in WordStar beim Starten voreingestellt ist, läßt sich per Patch ändern. Und auch das mitgelieferte Installations-Programm für den WordStar tut nichts anderes, als derartige Änderungen in die Datei WS.COM einzupatchen.

Lassen sich die genannten Voreinstellungen (zumindest bei den WS-Versionen ab 3.3) inzwischen fast alle per Installer auf maximalen User-Komfort trimmen, so sieht es beim Anpassen von Terminals schon schlechter aus. Besonders häßlich ist das bei den Versionen für den IBM PC. Hier läßt sich der Installer auch mit Tricks nicht dazu bringen, ins Terminal-Auswahl-Menü zu verzweigen. Will man also hier Manipulationen vornehmen, dann bleibt nur noch Patchen.

Dazu kann man natürlich auch einfach einen Debugger nehmen. Sollten Sie so etwas noch nie gemacht haben und von gewissen Selbstzweifeln geplagt sein ('Ich könnte ja was kaputt machen'), seien Sie beruhigt: Solange Sie niemals Ihre Original-Datei, womöglich auf der Master-Diskette, mit der neuesten Kreation überschreiben, kann auch nichts Unwiderrufliches passieren. Und wenn eine Version Seltsames macht oder einfach nur sang- und klanglos abstürzt – nun, dann wird halt eine neue gemacht. In diesem Sinne, frisch ans Werk.

Als Standard, auf den wir uns im folgenden beziehen, gilt der WordStar 3.0, dem wir uns in puncto Patching bereits in c't 1/84 ausführlich gewidmet haben. Dort sind auch die Bedeutungen der meisten Patches ausführlich erklärt. Diesen Beitrag finden Sie übrigens auch in unserem gerade erschienenen Sonderheft 'Software-Know-how' wieder.

Installation

Die drei zusätzlich zur Version 3.0 betrachteten WordStars sind die Version 3.3 für CP/M-80, die Version 3.35-1.0 für CP/M-86 und die Version 3.4 für PC-/MS-DOS. Alle drei neueren WordStar-Versionen sind mit einem Installations-Programm versehen, das aus zwei Modulen besteht, dem ausführbaren Teil (INSTALL.COM, .CMD oder .EXE, je nach Betriebssystem) und einer produktspezifischen Datei namens WS.INS. Die ausführbare Datei ist ein sogenannter General-Installer für alle möglichen MicroPro-Programme, erst mit der anderen Datei kann ein bestimmtes Produkt installiert werden.

Der Installer ist erheblich einfacher zu bedienen als bisherige Versionen, allerdings auch nervtötend langweilig mit seinen dauernden Rückfragen, ob die Eingabe nun auch wirklich korrekt ist. Anders als die früheren Installer bieten die neuen aber nicht mehr von sich aus an, den Benutzer in die heiligen Sphären der Patch-Area zu geleiten.

Label-Suche

Aber die Möglichkeit der sogenannten 'Customer-Modification' ist geblieben. In diesen Byte-Patch-Modus gelangt man durch Eingabe eines Pluszeichens ('+') anstelle der Buchstaben 'A' bis 'E' beziehungsweise 'X', die das Menü vorschlägt.

Interessant ist diese Betriebsart vor allem dadurch, daß man damit die absoluten Adressen aller Labels herausarbeiten kann. Hat man also nur die Unterlagen über die Patch-Area des WS 3.0, kann man sich die korrespondierenden Adressen für andere Versionen selbst ermitteln. Wesentlich ist ja nur, daß man auch weiß, was man wie mit den jeweiligen Speicherzellen beeinflussen kann. Und diese Möglichkeiten sind weitgehend gleich. Wie unsere Betrachtung allerdings zeigt, gibt es bei den Versionen für die 16-Bit-CPUs einige deutliche Abweichungen.

Um Mißverständnissen vorzubeugen: Es soll hier nicht der Raubkopierer mit Unterlagen versorgt werden. Wenn es Unterlagen gibt, dann hat dieser sie sowieso mitkopiert. Aber es gibt diverse spezielle Lieferversionen des WordStar, zum Beispiel die für den Osborne-Kit, wo diese nicht zum Standard-Lieferumfang gehören. Und bei denen kann man sich mit unserer Methode recht gut behelfen.

WS 3.3

Die auffälligste Neuerung beim Starten von WS 3.3 ist seine Meldung (Signon Message). Hier kommt erst ein WordStar-Logo und dann eine bildschirmfüllende Copyright-Notiz.

Das Inhaltsverzeichnis, das anschließend angezeigt wird, ist alphabetisch sortiert. Auch werden keine Dateien mit den Extensions '.COM', '.HEX', 'OVR' und '.$$$' angezeigt. Das ist zwar einerseits praktisch, weil das Inhaltsverzeichnis dadurch übersichtlicher wird, daß nur potentielle Textdateien angezeigt werden. Andererseits ist es unglücklich, wenn man Platz auf der Diskette schaffen muß, aber sich nicht von Textdateien trennen mag.

Weitere Änderungen fallen nicht so ins Auge. Beim Suchen und Tauschen ist eine leichte Geschwindigkeitssteigerung zu verzeichnen. Phantom-Spaces (^PF) werden jetzt als Zeichen mitgezählt und immer richtig angezeigt. Um die Microjustification ein-/auszuschalten, kann man jetzt .uj on beziehungsweise .uj off eingeben, ohne daß WordStar mit einem Fragezeichen am rechten Rand reagiert.

Eine wichtige Verbesserung ist, daß WordStar 3.3 prüft, ob ein File, das gerade benutzt wird, manipuliert werden soll. Bei WordStar 3.0 (und auch wieder bei 3.4) kann zum Beispiel folgender Fall eintreten:

Es soll nur eine Seite aus einer 100 Seiten langen Datei ausgedruckt werden. Wird diese eine Seite versehentlich unter gleichem Namen wie die Hauptdatei statt auf den Drucker auf die Diskette umgeleitet – dann ist

die Hauptdatei hinterher nur noch eine Seite lang. Das kann mit der Version 3.3 nicht passieren.

Überhaupt ist der WordStar 3.3 um einiges fehlertoleranter geworden, etwa bei vollem Directory. Es finden sich eine Reihe solcher Detail-Verbesserungen, recht kraß hat sich allerdings die Adreßlage der signifikanten Patch-Labels verändert. Hier sind teilweise wesentlich größere Abweichungen zu verzeichnen als bei den 8086/88-Versionen.

WS 3.35

Diese Version hat große Ähnlichkeit mit der Version 3.3. Das Directory ist auch alphabetisch sortiert, und Dateien mit den Extensions '.CMD,', '.H86', '.OVR' und '.$$$' werden nicht aufgeführt. Etwas inkonsequent hingegen die Tatsache, daß Systemdateien (Extension '.SYS'), also etwa das Betriebssystem CPM.SYS, angezeigt werden. '.uj on' wird wieder mit Fragezeichen quittiert, das versehentliche Überschreiben der aktuellen Arbeitsversion wird abgefangen.

Eine große Unzulänglichkeit der englischen Version 3.35-1.0, die uns vorlag, besteht darin, daß sie noch nicht auf die speziellen Umlaute des IBM PC eingerichtet ist. Man muß mit Klammern, Backslash und Schlangenlinie auf dem Bildschirm vorliebnehmen.

Erst bei der PC-DOS-Version 3.4 hat man den WordStar überlistet, mit diesen Zeichen, die jenseits 80h liegen, auch als Schriftzeichen umgehen zu können. Denn normalerweise benutzt WS ja das siebte Bit zur Kennzeichnung von Soft-Spaces und Soft-Returns.

Wer unter CP/M-86 mit dem Debugger (DDT86) gegen seinen WordStar vorgehen möchte, der sollte daran denken, daß Programm-Dateien unter diesem Betriebssystem mit einem 80h langen Header vorab gekennzeichnet sind. Dieser enthält Ladeinformationen für das Betriebssystem und gilt nicht als Bestandteil des Programmes.

Beispiel: Sie laden den WordStar mit dem DDT86 an die Adresse 1000:0000, also 10000h absolut.

DDT86 <CR>
RWS.CMD 1000:0 <CR>

Label	WordStar-Version				Funktion
	3.0	3.3	3.35	3.4	
HITE	248	232	248	248	Bildschirmhöhe
WID	249	233	249	249	Bildschirmbreite
CLEAD1	24A	234	24A	24A	Cursor-Positionierung Anfang
CLEAD2	24B	23D	253	253	„ (String zwischen Zeile und Spalte)
CTRAIL	258	242	258	258	„ (String nach Zeile und Spalte)
CB4LFG	25D	247	25D	25D	Flag, ob erst Zeile oder Spalte kommt
LINOFF	25E	248	25E	25E	Zeilenoffset
COLOFF	25F	249	25F	25F	Spaltenoffset
ERAEOL	26D	250	26D	26D	Löschen bis Zeilenende
LINDEL	274	257	274	274	Cursor-Zeile löschen
LININS	27B	25E	27B	27B	Zeile einfügen
IVON	284	267	284	284	Invers ein
IVOFF	28B	26E	28B	28B	Invers aus
TRMINI	292	275	292	292	Terminal-Initialisierung
TRMUNI	29B	27E	29B	29B	Terminal-Abschieds-String
MEMAPV	2B0	290	2B0	2B0	Terminal ist „memory-mapped"
MEMADR	2B1	291	2B1	2B1	Adresse des Bildspeichers
MORPAT	2E0	2CB	2E0	266	Bereich für eigene Routinen

Terminal-Patch-Area

Label	WordStar-Version				Funktion
	3.0	3.3	3.35	3.4	
USELST	2AA	28D	2AA	2AA	wenn letztes Zeichen auf Schirm verwendet werden kann, $FF eintragen
DELCUS	2AE	28E	2AE	2AE	Delay nach Cursor-Adressierung
DELMIS	2AF	28F	2AF	2AF	Delay nach anderen Funktionen
HIBIV	2B3	293	2B3	2B3	inverses Video durch oberstes Bit
DEL1	2CF	2AF	2CF	2B4	kurze Delays (Blink-Frequ.)
DEL2	2D0	2B0	2D0	2B5	mittel-kurze Delays (Blink-Frequ.)
DEL3	2D1	2B1	2D1	2B6	mittel-lange Delays (Menüs)
DEL4	2D2	2B2	2D2	2B7	lange Delays (Meldungen)
DEL5	2D3	2B3	2D3	2B8	Delay bei horizontalem Scrolling
DEFDSK	2DC	2B9	2DC	2C1	Default drive für Overlays (B = 2, C = 3 . . .)
SCRLSZ	2DD	2BA	2DD	2C2	Zeichenzahl, die horizontal gescrollt wird
ITHELP	360	34D	360	346	anfängliche Hilfsstufe
ITITOG	362	34F	362	348	Einfügen ein/aus
INITLM	37F	36A	37F	365	anfänglicher linker Rand (-1)
INITRM	380	36B	380	366	anfänglicher rechter Rand (-1)
NONDOC	392	378	392	378	Document- oder Nondocument-Modus
DECCHR	393	37A	393	379	Zeichen für Dezimal-Punkt (. oder ,)
HZONE	39A	3C9	39A	380	zum Trennen kurzer Worte erniedrigen (auf 1)
ITPOPN	3D3	3FD	3D3	4C3	anfänglich Seitennummern oder nicht

Patch-Area für Voreinstellungs-Flags

Label	WordStar-Version				Funktion
	3.0	3.3	3.35	3.4	
POSMTH	690	699	746	8E4	Methode zum Überdrucken
BLDSTR	691	69A	747	8E6	wie oft wird bei BOLDFACE gedruckt
DBLSTR	692	69B	748	8E7	wie oft wird bei DOUBLE STRIKE gedruckt
PSCRLF	696	69C	74C	8EB	CR/LF-String
PSCR	6A1	6A7	757	8FA	CR-String
PSHALF	6A8	6AE	75E	905	halber Linefeed
PBACKS	6AF	6B5	765	912	Backspace
PALT	6B5	6BB	76B	91F	zweite Schriftdichte
PSTD	6BA	6C0	770	92C	normale Schriftdichte
ROLUP	6BF	6C5	775	939	Hochrollen
ROLDOW	6C4	6CA	77A	953	Runterrollen
USR1	6C9	6CF	77F	96D	User 1 (was wird gesendet, wenn ↑PQ gefunden wird)
USR2	6CE	6D4	784	97A	User 2 (bei ↑PW)
USR3	6D3	6D9	789	987	User 3 (bei ↑PE)
USR4	6D8	6DE	78E	994	User 4 (bei ↑PR)
RIBBON	6DD	6E3	793	9AE	Rot-Umschaltung
RIBOFF	6E2	6E8	798	9BB	Rot aus
PSINIT	6E7	6ED	79D	9C8	Drucker-Initialisierung
PSFINI	6F8	6FE	7AE	9E9	Drucker Ende
CSWTCH	717	717	7C9	ACB	Welchen Drucker-Treiber nehmen?
HAVBSY	718	718	7CA	ACC	Busy-Test vorhanden
LIBSY	71A	719	-	ACE	Sprung zum Busy-Test
LISEND	71D	71C	-	ADF	Zeichen zum Drucker senden
LISINP	725	724	-	AEF	Zeichen vom Drucker lesen (für Protokoll)
PROTCL	786	778	879	B7B	verwendetes Protokoll (0 = keins)

Drucker-Patch-Area

Dann sollten Sie die Segmentadresse um 80h höher einstellen, indem Sie beispielsweise einen Hexdump mittels

D1008:0 <CR>

in die Wege leiten. Wenn Sie weitere Dumps nur durch Angabe des Offsets ausgeben lassen, also etwa mit

D248 <CR>

dann liegen die Label-Adressen parallel zu unserer Tabelle, Sie brauchen also nicht dauernd die Adreßlage umzurechnen. Es empfiehlt sich, immer zuerst nachzuschauen, ob man bei 248h (HITE) 18 oder 19h und bei 249h (WID) 50h findet. So weiß man, daß man sich richtig 'synchronisiert' hat.

Wenn man Unterlagen zu 8080-WordStars hat, dann findet man darin auch die Standard-Belegungen diverser Speicherzellen. Vor allem dort, wo man eigene Unterprogrammaufrufe einbinden darf, findet man zwei NOPS und einen RET-Befehl.

Bei der Version 3.35 differiert das ein bißchen. Natürlich ist der Opcode für NOP bei der 8086 90h, gelegentlich findet man aber nicht die Sequenz 90 90 C3, sondern 00 90 C3. Denn manche dieser Blind-Routinen werden immer angesprungen, bei anderen wird nur nachgeschaut, ob dort was anderes als 00 steht.

An allen Patch-Stellen, wo man drastische Abweichungen findet, sollte man erstmal keine Änderungen probieren. Denn einiges weicht doch erheblich ab. So etwa die Bedeutung der Funktionen IVON/IVOFF bei den Versionen 3.35 und 3.4. Bei den 8080-Versionen wird hier im allgemeinen ein String zur Attribut-Einstellung übergeben. Dabei gibt das erste Byte die String-Länge an. Anders bei den Versionen für einen IBM-PC-Bildschirm: bei denen steht hier unmittelbar das Attribut-Byte!

Man kommt also bei einigen Sachen nicht darum herum, das Geld für eine ergänzende Dokumentation locker zu machen, wenn man mit Analogie-Schlüssen nicht mehr weiterkommt, aber unbedingt bestimmte Features einbauen will.

WS 3.4

Die Version 3.4 lag uns nur in der Version für PC-DOS vor. Herumstochern mit dem Debugger in Installer und Programm offenbart aber, daß es diese Version auch für CP/M geben muß. Sie meldet sich gänzlich anders, vor allem ist die Terminal-Installation völlig auf die Verhältnisse des IBM PC abgestimmt. Blockgrafikzeichen werden in den Menüs verwendet, und die Funktionstastenbelegung ist stets in einer Statuszeile eingeblendet.

Interessant ist allerdings, daß hier wieder 'Rückschritte' zu verzeichnen sind. Das Directory ist zwar immer noch alphabetisch sortiert, enthält aber wieder alle Dateien. '.uj on' wird wieder mit einem Fragezeichen quittiert, und auch das Überschreiben der aktuellen Arbeitsdatei ist wieder möglich. Dennoch ist es aber die für PC-User angenehmste Version, unter anderem auch deshalb, weil sie mit den Umlauten des PC korrekt umgeht.

Einen Adreßversatz gibt es unter PC-DOS nicht, beim Arbeiten mit dem Debugger muß also kein 80h langer Header berücksichtigt werden.

Bei der Version 3.4 zeigt sich allerdings am deutlichsten, daß Analogie-Schlüsse zu alten Versionen mit besonderer Vorsicht zu genießen sind. Diverse Labels haben bei der Version 3.4 keine Bedeutung mehr, im Installer stehen sie allerdings drin, folglich akzeptiert dieser sie. So wird sowohl für HIBCUR als auch CRBLIV 2B3h geliefert. Die betriebssystem- und versionsübergreifenden Installer fangen solche Fehler nicht ab.

Überhaupt sollte man bei den Adressen zur Terminal-Installation höchst vorsichtig sein. Bei der Version 3.35 haben die Terminal-Strings, beispielsweise zur Cursor-Positionierung, eine Funktion, bei der Version 3.4 für den PC sind sie zwar vorhanden, werden aber bei Installation auf die IBM-Farbgrafikkarte vollkommen ignoriert.

Literatur

[1] Bob Hickey, Kaypro steps up its Software, Microcomputing 1983, Heft 12, S.70 bis 73

[2] Steve Rimmer, WordStar am Zeug geflickt, c't 1984, Heft 1, S. 60 bis 63

[3] MicroPro, WordStar Reference Manual, San Rafael, 1983

ct

MARFLOW NEWS

Jetzt bei uns erhältlich:

c't-180 (c't 2/86):

Achtbit-Power auf Europaplatine als **Leerplatine**

inkl. Monitor-EPROM und Sourcelisting ..... **DM 138,00**

oder Fertigkarte mit 64K ..... **DM 598,00**
mit 128K ..... **DM 649,00**
mit 264K ..... **DM 689,00**
mit 512K ..... **DM 749,00**

Für weitere Informationen über die c't-180-Platine bitte ausführliches Prospektmaterial anfordern!!!!

Ein Konzept setzt sich durch:

c't-86/c't-68-ECB

das erste echte 16-Bit-ECB-Bus-System in Europaformat dazu NEU! **c't-86-Kompiettdokumentation** (ca. 300 Seiten). Schutzgebühr DM 75,00 als Grundlagenlektüre für den Aufbau und Betrieb Ihres c't-86 bestens geeignet'!

Jetzt verfügbar:
CP/M 68K für c't-68-ECB **DM 985,00!**

Für weitere Informationen über Hard- und Software des c't-86 bitte ausführliches Prospektmaterial anfordern!!!!

Jeder Computer benötigt sie:

Tastaturen — natürlich von CHERRY

Gehäuse für Einbau von Terminalplatinen vorgesehen!

Unser Spektrum reicht von der einfachen ASCII-Tastatur bis zur IBM-kompatiblen PC-Tastatur.

Sonderangebot:

Cherry-Matrix-Tastatur, speziell für Terminal A oder B (mit 10 auf drei Ebenen frei programmierbaren Funktionstasten):

Tastatur und Gehäuse .... **DM 279,00!** Tastatur einzeln .... **DM 199,00!**

Für weitere Informationen über Cherry-Tastaturen bitte ausführliches Prospektmaterial anfordern!

Nach wie vor aktuell:

c't-Terminal-Karten

— unentbehrlich für Ihren Computer! —
in Vorbereitung: **c't-Terminal C**
Das Terminal im Europaformat mit parallelem Attributspeicher!!!

Für weitere Informationen über c't-Terminal-Karten bitte ausführliches Informationsmaterial anfordern!!!!

Auch ein Einstieg in die Maschinensprache-Ebene:

SET-65 unschlagbar mit CEPAC-65!

Der Steuerrechner-EPROMMER-Trainer mit CMOS-Ein-Platinen-Allzweck-Computer in Verbindung mit dem Handbuch 6502/C02-Maschinensprache von Christian Persson. Eine ideale Kombination!
Für weitere Informationen über SET-65 und CEPAC-65 bitte ausführliches Prospektmaterial anfordern!!!!

Neuheiten aus der c't:

Wir liefern EPROM-Bank für Atari ST, Universelles Parallel-Interface für den Atari ST, c't-Uhr und RAM-Disk für den ECB-Bus.

Des weiteren liefern wir:

ECB-I/O-Karte, Z-80-BUS-Monitor, ECB-Adapter für Schneider, c't-Druckerspooler (verschiedene Ausführungen), Europakarten-Netzteil usw.

Für unsere Industrie-Kunden:
Wir haben ein komplettes **VMEbus-Kartenprogramm**

NEU!!!

SBC 68008-Singleboard-Rechner

Für weitere Informationen über unser VME-bus-Kartenprogramm bitte ausführliches Prospektmaterial anfordern!!!!

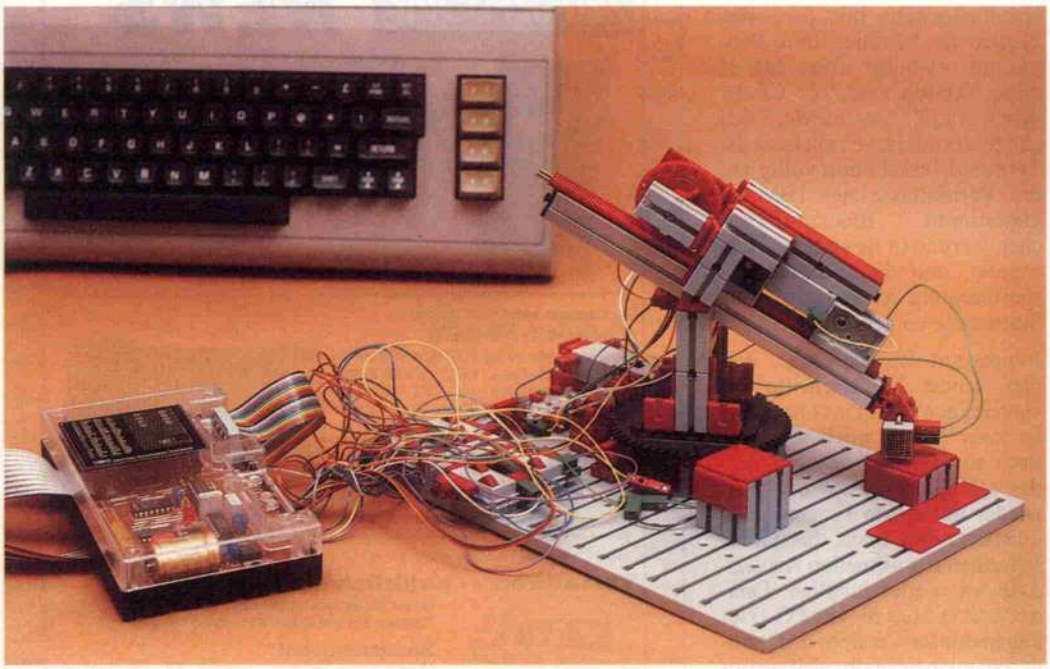
Manual einzeln Schutzgebühr DM 10,00/Karte.
Bei jeder Fertigkarte liegt entsprechendes Manual bei!
Auf allen Karten (aufgebaut und getestet) und Tastaturen 6 Monate Garantie.
Sämtliche Leerplatinen 100%ig elektronisch geprüft.
Alle Preise sind unverbindliche Richtpreise inkl. MwSt.
Mindestbestellwert DM 50,00!
Versand: per NN (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschkto. Hannover 1429 28-308, keine Versandkosten)

Technische Auskünfte freitags telefonisch zwischen 14.00—16.00 Uhr!!!!

oder fordern Sie gezielt unser ausführliches Prospektmaterial an!!!!

MARFLOW COMPUTING GmbH

Brüderstraße 2 · 3000 Hannover 1
Telefon 05 11/32 60 98



Fischertechnik Computing Interface

Lange Arme für Commodore

Eckart Steffens

Wer sich nicht auf dumpfe Bildschirmspielchen oder Zinssatzberechnungen in BASIC beschränken möchte, sondern überlegt, wie er seinen Heimcomputer möglicherweise doch noch irgendwie sinnvoll einsetzen kann, wird bald erkennen, daß eine Schnittstelle zur Kommunikation mit der Außenwelt fehlt. Schnittstelle, das bedeutet schalten und steuern, Impulse, Signale sowie analoge Werte aufnehmen und auswerten. Nicht nur eine Steuerung von zum Beispiel einer Modelleisenbahn, sondern bereits die Fernschaltung einer Kaffeemaschine bedingt ein solches Bindeglied.

Mit dem Computing Interface hat die von technisch orientierten Spielwaren her bekannte Firma Fischer nun eine solche Schnittstelle vorgestellt. Rechnerseitig ist ein Commodore-Computer mit 'Userport' Pflicht; es kann also sowohl ein CBM 3000/8000 als auch ein VC-20 oder ein C64 sein. Die kleinen Brüder C16/116 eignen sich mangels Userport-Anschluß nicht.

Was bietet nun das Interface? Zunächst einmal einige Schaltausgänge, die über Relaiskontakte potentialfrei realisiert sind. Damit lassen sich Motoren, Lampen oder Elektromagnete steuern, wie sie in den Fischertechnik-Baukästen oder einer Modelleisenbahnanlage vorkommen. In der umgekehrten Richtung, also zum Computer, bietet das Interface acht Signaleingänge für Schalter oder Taster (Endschalter, Umschalter) sowie zwei Analogeingänge, zum Beispiel für den Anschluß von Potentiometern. Der uns zur Verfügung gestellte Modellbaukasten enthielt auch solche Elemente – als Stellglieder zum Beispiel für die Rückmeldung von Motorstellungen (Antennenrotor).

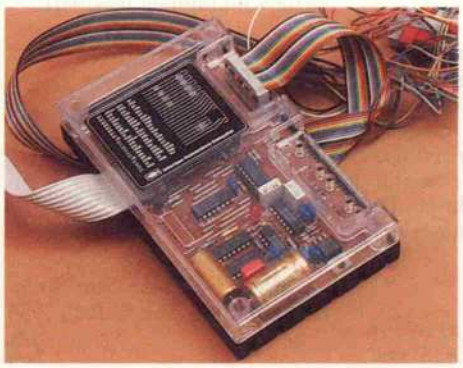
Der computergestützten Steuerung des selbstgebauten Miniroboters (unser Beispiel), der Eisenbahn oder der Autorennbahn steht also nichts mehr im Wege. Der Zusammenbau der Modellkastenelemente geschieht in der bekannten problemlosen Art und Weise; die neu geschaffenen Steuerelemente und Sensoren passen sich nahtlos in das System ein. Die Verdrahtung erfolgt mit den aus der Modellbahntechnik bekannten Miniatur-Bananensteckern und einfacher Litze. Will man nicht jedesmal die Leitungen nach Maß konfektionieren, läßt sich zwar ein 'Drahtverhau' nicht vermeiden – der Funktion tut das aber keinen Abbruch.

Das Interface ist in einem kleinen Kunststoffkästchen montiert, das mit einem Anschlußkabel für den Computer sowie den Buchsen für die Steuerausgänge und -eingänge ausgestattet ist. Während die Stromversorgung der Elektronik aus dem Computer erfolgt, ist für den Technik-Baukasten ein eigener Trafo zu verwenden: Computerseite und Modellbauseite sind elektrisch vollkommen getrennt, was Schäden vermeiden hilft.

Das Grundprogramm sowie eine Anzahl lauffähiger Programme, von der Verkehrsampel bis zum Roboter, liefert Fischer auf Diskette mit. Die Programme sind listbar, die Initialisierungs- und Systemdiagnoseroutine sogar als Listing zusätzlich abgedruckt. Dadurch kann man leicht nachvollziehen, wie die Programmierung angelegt ist, und auf den Beispielen eigene Anwendungen aufbauen. Es ist erfreulich, daß alle Programme sowohl für den VC-20 als auch für den C64 geschrieben wurden. Bei beiden Computern gab es keine Probleme, da sogar das Bildschirmformat von den Autoren der Software angepaßt wurde. Die Modifikationen für CBM 3000/8000-Rechner sind einfach und im Begleitmaterial beschrieben.

Ist das Initialisierungsprogramm geladen, kann man mit Befehlen wie 'SYS M1,EIN' oder 'SYS M1,RECHTS' zum Beispiel den Motor 1 einschalten und nach rechts laufen lassen. Spezial-Programmierkenntnisse sind also nicht erforderlich.

Fazit: Auch unter Nichtbeachtung des neu entflammten Spieltriebs ist das Fischertechnik-Computing-Interface eine gelungene Erweiterung, mit der sich für den Homecomputer ein völlig neues Betätigungsfeld eröffnet.



Mit dem Computing Interface von Fischertechnik kann man einen Computer sinnvoll einsetzen — zum Steuern von Modellen. c't

68020 Computer

12,5 MHz 68020 32-Bit Prozessor · 68881 Gleitkomma-Koprozessor optional · 2 MB RAM organisiert als 512 KB x 32 Bit · 256 KB EPROM max. mit 2764 / 27128 / 27256 / 27512 · 4 x serielle Schnittstellen · 8-Bit Parallelschnittstelle · Erweiterungsanschluß für Ein-/Ausgabe · Datum, Uhrzeit Batterie gepuffert · 5" Floppy-Kontroller · SASI Interface für intelligente Winchester Laufwerke · Programmierbarer Interrupt Generator · Hardware Single Step Logic · Abmessungen 100 x 140 mm

Betriebssystem OS-9/68K
oder Motorola 020Bug

System mit 5" Floppy, 20 MB Winchester ab DM 19 999 incl. MwSt.

ZACHER

Zacher Computer GmbH · Im Schwarzenstein 34 · 5521 Irrel
Tel. 06525/299 · Telex 4729608dzi

GAS
GESELLSCHAFT FÜR AUTOMATION UND SYSTEMENTWICKLUNG mbH
AACHENER STR. 11 · 5000 KÖLN 90 · 02203/291311

ARTIFICIAL INTELLIGENCE EXPERTEN-SYSTEME

EXPERTEN SYSTEM Entwicklungs Pakete

EXPERT EASE
Hardware: IBM PC min. 128 kB und eine 5¼" Floppy
System: entwickelt in UCSD Pascal
Preis: 2200,— DM
Support: Handbuch deutsch

EXSYS
Hardware: IBM PC min. 256 kB und eine 5¼" Floppy
System: entwickelt in C
Preis: 1200,— DM
Support: Handbuch englisch

ES/P Advisor
Hardware: IBM PC min. 256 kB und eine 5¼" Floppy
System: entwickelt in Prolog
Preis: 2200,— DM
Support: Handbuch englisch

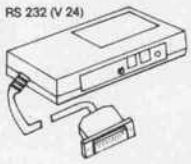
EXPER-OPS5
Hardware: MACINTOSH min. 256 kB Floppy
System: entwickelt in Lisp
Preis: 2500,— DM
Support: Handbuch englisch

G.A.S. EXPERTEN SYSTEM
Hardware: IBM PC min. 128 kB und eine 5¼" Floppy Texas PC, alle IBM-PC compatible
System: entwickelt in BASIC
Preis: 350,— DM (Source Files)
Support: Beschreibung deutsch

NEW MODEM

C64
DM 138,-

RS 232 (V 24)



IBM
DM 198,-

resco electronic

Hardware
* 300 Baud
* Voll-Halbduplex
* Originate/Answer
* CCITT V. 21 (deutsche Norm)
* Wählautomatik
* Automatische Rufannahme (Mail-Box-Betrieb!)
* Telefon/Modem-Umschalter
* LED für Empfangsanzeige

Software
* Voll menü-gesteuert
* Automatischer Verbindungsaufbau
* Rufnummernspeicher
* Daten- und Programmübertragung
* 30-kB-Pufferspeicher
* ON-Line-Uhr
* ASCII oder Commodore-Mode (C 64)

Software für C64 DM 48,-
dto. in Deutsch mit Wahlwiederholung DM 68,-
Mailbox-Programm für C64 DM 99,-
Software für IBM (Preise inkl. MwSt.) DM 98,-

Zahlung per Nachnahme oder Vorausscheck!

Momentan nur ohne fernmelderechtliche Genehmigung erhältlich.

Händleranfragen erwünscht.

Hessenbachstrasse 35
D-8900 Augsburg
Telefon 08 21/52 40 33
Telex 53 776 resco d

c't-Platinen

c't-Platinen bestehen aus Epoxid-Glashartgewebe, sind fertig gebohrt und mit Lötstopplack versehen bzw. verzinnt. Weitere Merkmale können Sie der Buchstabenkombination in der Bestellnummer entnehmen; die Buchstaben bedeuten: 'd' — doppelseitig, 'B' — Bestückungsaufdruck, 'E' — elektronisch geprüft.

Nr.	Projekt	Format	Preis
c't 86/c't 68 EBC			
840150d	Busplatine (96pol., 10 Steckplätze)	84 x 208 mm	49 DM
840147dBE	CPU-Karte	Europa	85 DM
840149dBE	I/O-Karte, inkl. PROM	Europa	79 DM
840288dBE	Floppy-Interface, inkl. PROM	Europa	75 DM
850164dBE	RAM-Karte 1 MByte, inkl. PROM (bei Bestellung Speicher-Konfiguration angeben)	Europa	98 DM
850584dBE	Farbgrafikkarte	Europa	98 DM
	Farbgrafikkarte inkl. EPROM und 6 PALs		298 DM
850870dBE	IFC-Karte mit 3 PALs und EPROM	Europa	198 DM
851098dBE	Unicard — Universelle Erweiterungskarte inkl. PROM	Europa	89 DM
851162dBE	68000-CPU-Karte inkl. PAL und 2 EPROMs	Europa	198 DM
c't 80			
	Platinen für den c't 80-Computer werden grundsätzlich nur inklusive Firmware (EPROMs) geliefert		
840652dBE	Grafik-Interface GRIP (EC8-Bus) Platine mit Betriebsprogramm-EPROM 27128 und Programmbeschreibung	Europa	245 DM
840826dBE	PROF-80 (CPU/RAM/Floppy-IF), Platine, Monitor-EPROM, Assembler-Listing	Europa	178 DM
	PROF-80-Platine mit 6-MHZ-EPROM und Listing (Listing und Firmware des Monitorprogramms weichen zum Teil voneinander ab, weil die Firmware weiterentwickelt worden ist. Ein Listing, das dem neuesten Software-Standard entspricht, ist leider nicht lieferbar.)	Europa	188 DM
ECB-Boards			
840184d	CEPAC-80 B (mit Wrap-Feld)	Europa	69 DM
840187d	CEPAC-80 A (ohne Wrap-Feld)	ca. 86 x 100 mm	49 DM
840782dB	EPAC-80 A (ohne Wrap-Feld)	ca. 80 x 100 mm	39 DM
840783dB	EPAC-80 B (mit Wrap-Feld)	Europa	59 DM
850294dB	PROMMER-80 inkl. Platine für Programmiersocket (80 x 25 mm)	Europa	69 DM
850484dB	I/O-Karte	Europa	79 DM
851074dB	ECB-Busmonitor	Europa	69 DM
860230dBE	c't 180, CPU-Karte inkl. Monitor-EPROM und Source Listing (lieferbar ab März)	Europa	138 DM
Klang-Computer			
841242B	ADS-Vorverstärker und ADS-Slotkarte	ca. 104 x 47 mm ca. 112 x 80 mm	38 DM
850138B	DSM	ca. 140 x 68 mm	15 DM
850252dBE	KBI-Slotkarte	ca. 77 x 160 mm	39 DM
850386B	KBC-Karte	ca. 210 x 45 mm	22 DM
850387B	KBB-Karte	ca. 220 x 75 mm	27 DM
850388B	KBE-Karte	ca. 220 x 75 mm	21 DM
850389B	1 Satz aus 1 x KBB und 3 x KBE		85 DM
850450dBE	PCS-Slotkarte	160 x 77 mm	42 DM
8506124dB	Voice RAM	ca. 150 x 160 mm	49 DM
841243	Satz aus 8 Voice-RAM-Karten		369 DM
841244	Kompletter Kartensatz für Maximalausbau (ADS-Vorverstärker, ADS-Slotkarte, KBI-Slotkarte, KBC, UBB, 3x KBE, PCS, 8x Voice RAM) inkl. Programmdiskette		598 DM

Nr.	Projekt	Format	Preis
c't 68000			
	Platinen für den c't68000-Computer werden grundsätzlich nur inklusive Firmware (EPROMs, PALs, PROMs) geliefert		
841167dBE	Europakartenversion (Leerplatinensatz aus CPU, Switchboard, I/O-FDC, Peripherieadapter, DRAM, SBI-EBCS, inkl. MIKROMON, RTOS, PEARL-Compiler in EPROMs. Dekoder-PALs, Handbuch, jedoch ohne Grafikkarte, Bus-Monitor, Backplane)	Europa	672,60 DM
850190dBE	Grafikdisplay-Prozessor, Leerplatine inkl. PAL	Europa	108,30 DM
841168dB	Busmonitor-Karte (inkl. PROMs)	Europa	62,70 DM
850663dB	Farbgrafik-Erweiterungskarte	Europa	96,90 DM
831241dBE	Terminal A (ohne Tastatur)	ca. 84 x 234 mm	59 DM
831242dBE	Terminal B (mit Tastatur)	Doppel-Europa	75 DM
831262	Universelles Netzteil	Europa	18 DM
840242B	Centronics/V24-Interface für Olympia COMPACT	80 x 136 mm	15 DM
840252B	c't-Sprachsynthesizer	100 x 117 mm	21 DM
840352dB	CEPAC-65, Version A	80 x 100 mm	27 DM
840354dB	CEPAC-65, Version B	Europa	52 DM
840496dB	PIO-Drucker-Interface für ZX 81	Europa	30 DM
840529d	PIO-Drucker-Interface für ZX Spectrum	Europa	30 DM
840536	ScopeExtender (Rückseite mit Frontplattenaufdruck)	ca. 78 x 148 mm	19 DM
840538	Netzteil für ScopeExtender (±5 V, 3,3 VA)	78 x 148 mm	8 DM
840726dB	SET-65 (Ergänzungsplatine)	100 x 183 mm	32 DM
840727	Tastensatz und Display-Aufkleber (bedruckt) für SET-65		27 DM
841051dB	Schrittmotorsteuerung	ca. 63 x 190 mm	30 DM
841056dB	Soft-ROM für ZX 81	ca. 50 x 100 mm	24 DM
8412112dB	EPROM-Bank für C64	ca. 80 x 58 mm	18 DM
850469dB	C64-Logikanalysator-Zusatz	100 x 150 mm	49 DM
850170dB	C64-Speicheroszilloskop-Zusatz	ca. 100 x 150 mm	49 DM
850346dBE	EPAC 95 A (ohne Wrap-Feld)	ca. 90 x 100 mm	45 DM
8505100dB	SuperTape-Interface für TRS-80	ca. 73 x 39 mm	18 DM
850584B	Videoentzerrer	ca. 94 x 58 mm	12 DM
850570dB	Programmierbarer EPROM-Simulator PEPS	ca. 70 x 110 mm	48 DM
850676dBE	Drucker-Spooler	ca. 138 x 74 mm	49 DM
850680B	X-Schalter	ca. 100 x 120 mm	27 DM
850772d	96pol. Bus-Extender	ca. 100 x 240 mm	55 DM
850667	Steckplatzadapter ROM/EPROM	ca. 23 x 37 mm	3 DM
850779	Steckplatzadapter ROM/EPROM (doppelt)	ca. 35 x 45 mm	6 DM
8506116B	Spectrum-NMI-Karte	ca. 85 x 90 mm	14 DM
850774dB	IEC-Interface für C64	ca. 58 x 72 mm	18 DM
850888dB	8"-Controller für Apple II, Slotkarte, Kontakte vergoldet	ca. 84 x 76 mm	33 DM
850958dB	Kompaktnetzteil (4 Spannungen)	Europa	42 DM
8510110dB	32 • I/O-Slotkarte für Apple, Kontakte vergoldet	ca. 82 x 78 mm	28 DM
851082dBE	68000-Busmonitor	Europa	69 DM
851254dBE	ECB-Adapter für Schneider CPC	Europa	59 DM
860158dB	EPROM-Bank für Atari ST, Steckkarte, Kontakte vergoldet	ca. 56 x 128 mm	29 DM
860360dB	I/O-Karte (User-Port) für den Atari ST, Kontakte vergoldet, mit 2 Steckplätzen für EPROMs	ca. 72 x 179 mm	49 DM
860361	Programmiertes PAL dazu		29 DM
860444dB	c't-Uhr inkl. PAL	ca. 52 x 60 mm	53 DM

Bitte beachten Sie: Alle in der Liste aufgeführten Leerplatinen stehen im Zusammenhang mit Projekten der Zeitschrift c't. Die zum Aufbau erforderlichen Angaben sind der veröffentlichten Projekt-Beschreibung zu entnehmen. Zusätzliche Informationsschriften sind nicht erhältlich. Eine Fotokopie der Veröffentlichung können Sie unter Angabe der Platinennummer bestellen. Jede Fotokopie eines Beitrags kostet 5 DM, unabhängig vom Umfang. Das Platinenlayout entspricht jeweils der veröffentlichten Schaltung; Änderungen, insbesondere Verbesserungen, behalten wir uns vor. Solche Änderungen werden dann in geeigneter Weise dokumentiert, in der Regel durch Veröffentlichung in der Rubrik 'Ergänzungen + Berichtigungen'. Eine Gewähr für das fehlerfreie Funktionieren der betreffenden Schaltung kann jedoch nicht übernommen werden.

Ihre Bestellung richten Sie bitte an:

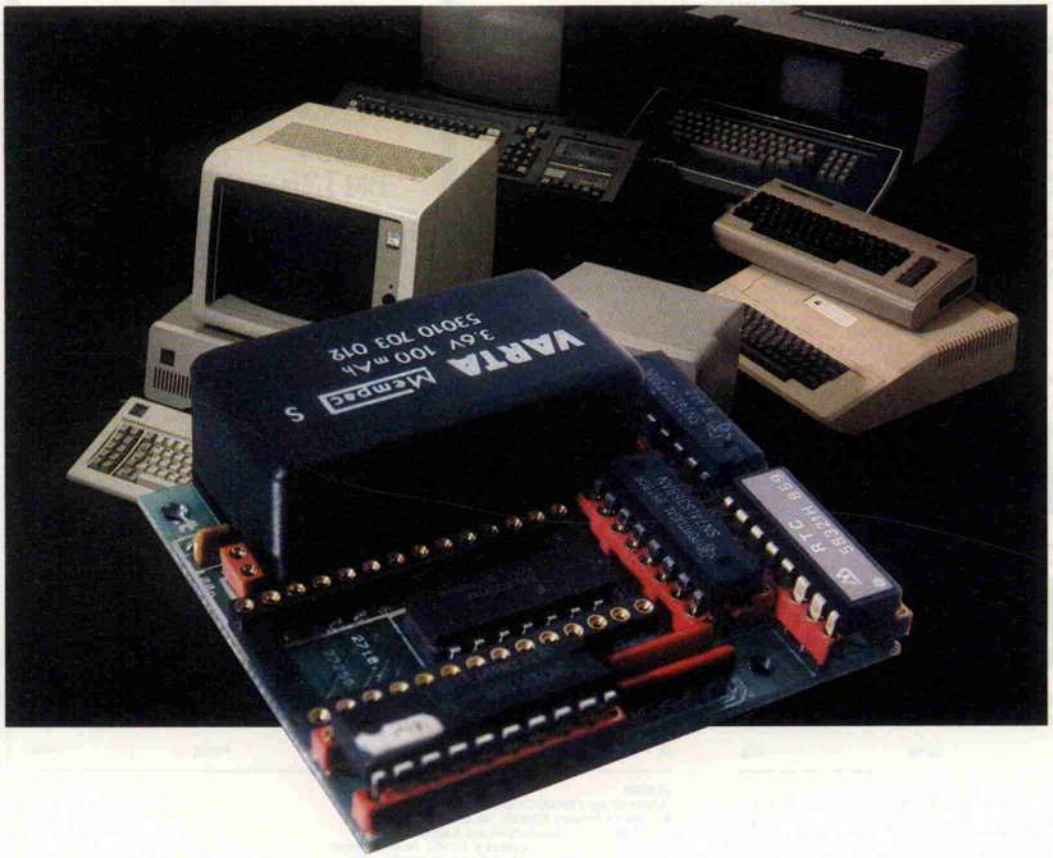
So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck über die Bestellsumme zuzüglich DM 3,— (für Porto und Verpackung) bei oder überweisen Sie den Betrag auf eines unserer Konten.

Schecks werden erst bei Lieferung eingelöst. Wir empfehlen deshalb diesen Zahlungsweg, da in Einzelfällen längere Lieferzeiten auftreten können.

Bankverbindungen:
Postscheckamt Hannover, Kt.-Nr. 93 05-308
Kreissparkasse Hannover, Kt.-Nr. 000-019968 (BLZ 250 502 99)

c't-Versand, Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61



Die versteckte Uhr

Akkugepufferte Echtzeituhr für jeden Rechner

Eberhard Meyer

Eine Echtzeituhr mit Akkupufferung fehlt bisher in vielen Rechnern. Das muß nicht so bleiben. Die c't-Uhr bietet von Sekunden bis zur Jahreszahl alle Daten, die Ihr PC braucht. Eine Schaltjahreskorrektur ist Ehrensache. Die kleine, feine Schaltung läßt sich in praktisch jedem Rechner einsetzen und stellt damit unseres Wissens ein absolutes Novum dar. Allenfalls die Bauhöhe der kleinen Zusatzplatine könnte dem Einbau im Wege stehen, doch auch da gibt es Kniffe...

Mal ehrlich: Wären Sie vor der Lektüre dieses Beitrags gefragt worden, ob man eine Schaltung entwickeln könne, die sich ohne Änderung in jeden Computer einstecken läßt, Sie hätten geantwortet: 'Ausgeschlossen!' Zu unterschiedlich ist die Hardware des unüberschaubaren Angebots an Computermodellen. Die Erweiterungssteckplätze des Apple beispielsweise sind völlig inkompatibel zu denen des IBM PC. Und viele Mikrocomputer bieten noch nicht einmal die Möglichkeit, ihre Hardware zu erweitern.

Und es geht doch! Denn eines haben sie alle gemeinsam: Irgendwo im Rechner steckt ein ROM, EPROM oder wenigstens ein Bytewide-RAM (organisiert in 8-Bit-Breite). Das gilt für den C64 ebenso wie für den Atari ST und den IBM PC. Wenn man also eine universelle Schaltung bauen will, so braucht man ihr nur einen Steckanschluß mitzugeben, der in einen EPROM/ROM-Sockel paßt. Da dieser Anschluß nicht nur genormt ist, sondern auch die Versorgungsspannung gleich mitliefert, scheint diese Lösung ideal. Sie deckt die gesamte Spannweite vom Einplatinenrechner bis zur Mainframe ab. Lediglich Kleinstrechner mit miniaturisierten Bauelementen fallen hier aus dem Rahmen.

Ein paar Probleme gibt es dennoch zu bedenken: Der EPROM-Sockel ist in der Regel nicht leer; es steckt ein Speicherbaustein drin, den der Rechner bis auf das letzte Byte ausliest. Für die Zusatzschaltung bleibt also eigentlich gar kein Adreßraum übrig. Außerdem können EPROMs nur gelesen werden; eine Steuerleitung, die auch das Schreiben von Daten ermöglicht, ist an dem Sockel nicht vorhanden. Außerdem gibt es recht unterschiedliche Typen von Bytewide-Speichern. Allein acht EPROM-Typen mit zwei unterschiedlichen Gehäusebauformen müssen unterstützt werden; dazu kommen ROM- und RAM-Bausteine mit ähnlicher Pinbelegung.

EPROM angezapft

Die Uhrenschaltung ist auf einer Huckepack-Platine aufgebaut, auf der das EPROM, in dessen Sockel sie eingesteckt wird, Platz findet. Fast alle Anschlüsse des EPROMs werden direkt zum Sockel auf der Rechner-Platine durchgeschleift. Lediglich das Output-Enable-Signal ($\overline{OE}$) kann von der Uhrenlogik unterdrückt werden.

Normalerweise wird jedoch auch $\overline{OE}$ direkt an den Festwertspeicher weitergeleitet. Herkömmliche Programme können die Existenz der Uhr also gar nicht erkennen. Jedes Byte des EPROMs kann nach wie vor ausgelesen werden. Die Logik der Uhrenkarte wird erst aktiv, wenn dreimal nacheinander die höchste Adresse des EPROMs angesprochen wird. Diese Zugriffe müssen unmittelbar hintereinander erfolgen. Wenn zwischendurch eine andere Adresse des EPROMs angesprochen wird, löscht dies den Zähler in der Zugriffslogik der Uhrenkarte. Mit dieser strengen Bevorrechtigung des Festwertspeichers wird sichergestellt, daß ein Programm nicht versehentlich den Zugriff auf die Uhr einschalten kann. Software, die im 'gelifteten' EPROM läuft, kann die Uhr grundsätzlich nicht ansprechen, denn jedes Lesen eines Opcodes führt zu einem Reset der Zugriffslogik. Mit einem kleinen Trick ist es aber dennoch möglich, die Uhr per EPROM auszulesen; dazu später mehr.

Nach dreimaligem Zugriff auf die höchste Adresse öffnet sich ein 128 Byte kleines Fenster ganz oben im EPROM-Adreßraum. Über dieses hat der Rechner nun Zugriff auf die Register der Uhr. Wenn eine Adresse außerhalb dieses Bereichs angesprochen wird, ist die Uhr wieder verschwunden.

Nr.	Funktion des Registers
0	Sekunden Einer
1	Sekunden Zehner
2	Minuten Einer
3	Minuten Zehner
4	Stunden Einer
5	Stunden Zehner, wobei Bit 3 = 1 für 24h-Zähler
6	Tag der Woche (Mo = 1)
7	Tag Einer
8	Tag Zehner
9	Monat Einer
A	Monat Zehner
B	Jahr Einer
C	Jahr Zehner
D	Reset der Sek.-Vorteiler bei WRITE
E & F	Taktsignale ausgeben, bei READ: D0 > 1024 Hz D1 > 1 Hz D2 > 1/60 Hz (Min.-Takt) D3 > 1/3600 Hz (Std.-Takt)

Register des Uhrenchips RTC58321

Als Uhrenbaustein kommt der RTC58321 zum Einsatz, den wir schon auf der 'Unicard' für den c't86 (Heft 10/85) verwendet haben. Er hat den Quarz bereits eingebaut, was hier besonders wichtig ist, da wir um jeden Quadratmillimeter Platz auf der Platine feilschen mußten.

Der RTC58321 verfügt über einen vier Bit breiten Datenpfad. Über diesen kann die Adresse eines internen Registers eingegeben werden. Dadurch selektiert man einen der 13 Zähler, die im BCD-Format (Binary coded decimal – binär kodierte Dezimalzahl) jeweils eine Stelle der Uhrzeit und des Datums wiedergeben. Die Sekunden-Einer haben die Adresse 0, die Sekunden-Zehner die Adresse 1 und so weiter (siehe Tabelle). Nachdem das Register ausgewählt ist, kann die betreffende Stelle nun ausgelesen oder beschrieben werden.

Beschrieben? Unsere Uhrenkarte sitzt doch auf einem EPROM-Sockel, der das Schreiben von Daten gar nicht erlaubt. Irrtum: Beim Zugriff auf das EPROM legt der Prozessor die Adressen an die entsprechenden Pins. Auch das sind Daten! Wer in der c't-Ausgabe 3/86 den Beitrag über den ST-Userport gelesen hat, erkennt hier die Parallele: Adreßleitungen werden zu Datenleitungen 'umfunktioniert'. Beim Zugriff auf einen bestimmten Block von 16 Adressen akzeptiert die Uhrenkarte das niederwertige Nibble als Datum und übernimmt dieses in einen Zwischenspeicher (74LS173).

Schritt	Lesen von Adresse	Kommentar
1	ffFF ffFF ffFF	Höchste Adresse des EPROMs dreimal lesen, um Uhrzugriff zu aktivieren (ff = 8..FF, je nach EPROM-Größe)
2	ffCn ffFn	Das Uhr-Register mit der Nummer n anwählen (siehe Tabelle)
3	ffE0 ffE0	Daten lesen. Beim 2. Lesen: untere 4 Datenbits = Uhr-Register
(4)	ffCF ffFF	(Nur zum Schutz gegen versehentliches Verstellen)
5	ff00 - 1	Uhrzugriff wieder abschalten

So wird die Uhrenkarte ausgelesen.

Schritt	Lesen von Adresse	Kommentar
1	ffFF ffFF ffFF	Höchste Adresse des EPROMs dreimal lesen, um Uhrzugriff zu aktivieren (ff = 8..FF, je nach EPROM-Größe)
2	ff8n ffBn	Das Uhr-Register mit der Nummer n anwählen (siehe Tabelle)
3	ff9d	Den Wert d einschreiben
(4)	ff8F ffFF	(Nur zum Schutz gegen versehentliches Verstellen)
5	ff00 - 1	Uhrzugriff wieder abschalten

So wird ein Register der Uhr gestellt.

Der Zwischenspeicher erscheint auf den ersten Blick unnötig. Man hätte ja auch die Daten direkt zum Uhrenchip durchgeben können. Der RTC58321 ist jedoch ein auf geringsten Stromverbrauch getrimmtes CMOS-IC, bei dem die Geschwindigkeit auf der Strecke geblieben ist. Laut Datenblatt muß man ihm beim Schreiben von Daten ganze 2µs Zeit lassen, damit er sich erst einmal an die vielen Einsen und Nullen gewöhnen kann. Deshalb sind alle Leitungen der Uhrenkarte, die zum RTC58321 führen, über Zwischenspeicher geführt. Die CPU kann jede Leitung setzen oder rücksetzen und hat so die absolute Kontrolle über die Uhr.

Für das Uhren-IC laufen die Zugriffe wie folgt ab: Zuerst wird die Adresse eingegeben und der AWRT-Pin für mindestens 0,5µs auf logisch 1 gepulst. Die Adresse braucht nicht vor dem AWRT-Puls angelegt zu werden, da das IC mit R/S-Flipflops an den Eingängen ausgerüstet ist. Nach der fallenden Flanke des AWRT-Signals jedoch muß die Adresse für mindestens 0,1µs stabil bleiben. Da nun eine Stelle in der Zählerkette ausgewählt ist, kann gelesen oder geschrieben werden. Zum Lesen wird einfach der READ-Pin auf logisch 1 gelegt. Der RTC58321 treibt daraufhin die Datenleitungen, und nach einer Mikrosekunde kann der Zählerstand von den Datenleitungen abgegriffen werden. Zum Schreiben werden die Daten angelegt und der WRITE-Pin auf logisch 1 gesetzt. Auch hier ist ein Vorlauf der Daten nicht erforderlich. Nach frühestens 2µs dürfen das WRITE-Signal und die Daten (hier ohne Nachlauf) wieder entfernt werden.

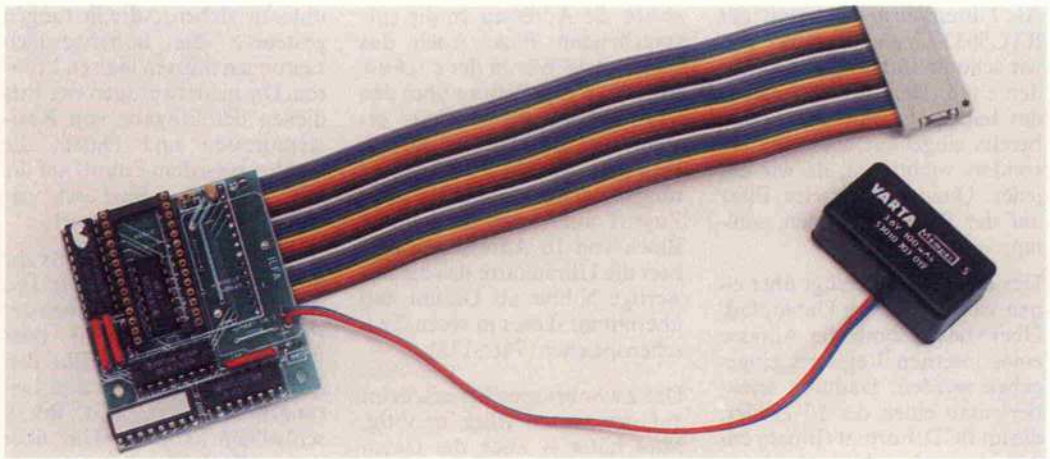
Wie sieht der Zugriff nun aus der Sicht des Programmierers aus? – Die Uhr wird über die unteren sieben Adreßleitungen gesteuert; die höherwertigen Leitungen müssen logisch 1 führen. Die niederwertigen vier Bits dienen der Eingabe von Registeradressen und Daten. Sie werden bei jedem Zugriff auf die Uhr in dem 4-Bit-Latch der Karte zwischengespeichert.

Bleiben noch drei Adreßbits zur Steuerung der Uhrenlogik: Die Bits 4 und 5 selektieren einen der Pins AWRT, WRITE oder READ. Haben beide Bits den Wert 1, wird keine der drei Leitungen aktiviert. Mit Bit 6 schließlich kann die Uhr über ihren STOP-Eingang angehalten werden. Falls diese Leitung bei Uhrenzugriffen logisch 1 führt, unterdrückt der RTC58321 intern für seinen Sekundentakt. Bei jedem Zugriff auf das angezapfte EPROM außerhalb des Uhrenadreßbereichs wird der STOP-Befehl automatisch wieder zurückgenommen.

Dicht gepackt

Die ICs liegen auf der Uhrenkarte recht dicht. Man sollte deshalb die folgende Reihenfolge beim Aufbau einhalten, um sich die Arbeit zu erleichtern: Als erstes Bauteil wird der IC-Stecker auf der Lötseite eingesetzt, der später in den EPROM-Sockel gesteckt wird. Leider sind diese Stecker schwer zu beschaffen und sündhaft teuer. Herkömmliche Stifte von Pfostensteckern sollten Sie aber auf keinen Fall nehmen, denn die sind zu dick und könnten den EPROM-Sockel im Rechner beschädigen. Man kann sich jedoch mit einer gedrehten IC-Fassung behelfen, deren Beine man als Stecker mißbraucht. In diesem Fall lötet man in jede der 28 Buchsen ein kurzes Drahtstückchen. Diese Drähte müssen nun von unten durch die Platine gesteckt und verlötet werden. Der Sockel darf dabei nicht ganz an der Platine anliegen, damit die Beinchen weit genug herausstehen und später einen guten Kontakt im EPROM-Sockel haben. Die nun auf der Bestückungsseite der Platine herausstehenden Drahtenden müssen sehr kurz abgekniffen werden, damit sie sich nicht an anderen Bauteilen stoßen.

Wenn es aus Platzgründen nicht möglich ist, die Uhrenplatine direkt in den EPROM-Sockel zu stecken, kann man sie statt



Bei engen Platzverhältnissen kann man den Akku oder die gesamte Platine über Kabel anschließen.

dessen mit einem 28poligen Flachbandkabel anschließen, das man erst zum Schluß der Bestückung anbringt. Das Kabel wird auf der Seite der Uhrenkarte angelötet, an die andere Seite kommt ein handelsüblicher Anpreßstecker. Irgendwo im Rechner findet sich sicher ein Plätzchen für die Uhr. Je nach Rechnertyp darf das Flachbandkabel jedoch nicht länger als ein paar -zig Zentimeter sein, damit keine Störungen auftreten.

Nur ein einziges IC muß gesockelt werden: das EPROM, das später dem Rechner entnommen und auf die Uhrenkarte gesteckt wird. Alle übrigen ICs können direkt eingelötet werden, was die Bauhöhe am Rand beträchtlich reduziert. Ein übriges kann man tun, indem man noch vor dem Griff zum Lötkolben die Beinchen auf der Lötseite abkneift. In den durchkontaktierten Löchern der Platine finden die ICs ausreichenden Halt; mit dem Seitenschneider sollte man nicht allzu zimperlich sein. Selbst wenn im Innern des Rechnergehäuses genügend Platz zur Verfügung steht, empfiehlt es sich, die Uhrenkarte so flach wie möglich aufzubauen; denn falls Sie irgendwann auf ein anderes Computermodell umsteigen, können Sie diese Hardware-Ergänzung sicher mitnehmen.

Statt des 74ALS30 kann in der Regel ein IC desselben Typs in LS-Ausführung eingesetzt werden. Der ALS-Chip ist um ein paar Nanosekunden schneller und bietet nur dann Vorteile, wenn das CS-Signal des Rechners sehr schnell auf die Adressen folgt. Wir fanden im c't-Labor nur einen 6502-Computer, bei dem dies tatsächlich kritisch war.

Falls die Schottky-Diode BYS21 oder ein ähnlicher Typ schwer zu beschaffen ist, können Sie auch eine gewöhnliche Siliziumdiode einsetzen, zum Beispiel die gute alte 1N4148.

Der Akku wird dann jedoch langsamer geladen, was zu Problemen führt, wenn der Rechner nur sehr selten eingeschaltet wird. Ein voller Akku gibt der Uhr eine Gangreserve von mindestens vier Monaten, in der Regel sogar wesentlich mehr.

Wenn die Karte vollständig aufgebaut ist, muß sie zum Schluß noch auf das verwendete EPROM eingestellt werden. Wer EPROMs der Typen 2716 oder 2732 verwendet, braucht sich darum nicht zu kümmern. Wenn ein EPROM vom Typ 2764. . .27256 angezapft werden soll, müssen Sie Ihr Taschenmesser zücken und auf der Lötseite eine dünne Leiterbahn unterbrechen, die die beiden Anschlüsse des Jumpers J verbindet. J liegt am unteren Rand des Akkus unmittelbar neben dem Blockkondensator C.

Falls die Uhr mit EPROMs der Typenreihe 25.. zusammenarbeiten soll, muß die Verbindung bei J bestehen bleiben. Statt dessen muß die Lötbrücke LBr geändert werden, die auf der Lötseite innerhalb der EPROM-Fassung zu finden ist. Die dünne Leiterbahn rechts der Lötbrücke wird aufgetrennt, und die untere Kupferfläche wird mit dem mittleren Steg durch einen dicken Zinnklecks verbunden.

Gelegentlich kommt es vor, daß die beiden Pins $\overline{OE}$ und $\overline{CS}$ der 27er- EPROMs im Rechner wie zwei NOR-Eingänge geschaltet sind. Auch in diesen Fällen wird die beschriebene Änderung der Lötbrücke erforderlich.

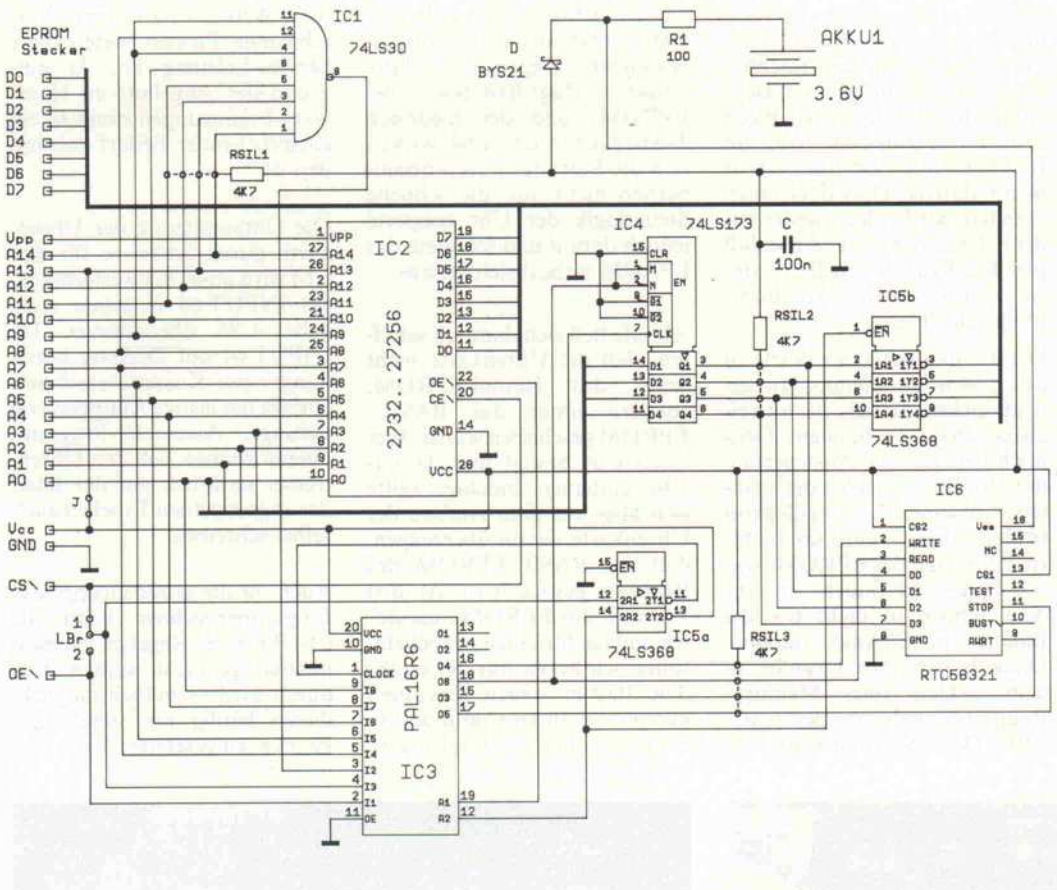
Wie spät ist es?

Die c't-Uhr ist per Programm sehr leicht zu handhaben. Die abgedruckten Programme zeigen, wie man sie in Z80-Assembler, in BASIC und in C bedient. Die Assembler-Routinen erfüllen die Konventionen für FORTRAN-Unterprogramme und können direkt in die FORLIB eingefügt werden. Dann ist es mit sehr einfachen Unterprogrammaufrufen von FORTRAN aus möglich, die Uhr auszulesen oder zu stellen.

Wer sich mit einem Debugger auskennt, kann die Routinen auch an beliebige lauffähige Programme anfügen. Dem M80 zum Beispiel bringt man so bei, stets das aktuelle Datum in das Listing zu drucken.

```
PAL16R6
Steuerschaltwerk für c't Universaluhr     e.m.2/86
Standardversion (EPROM- Zugriff grundsätzlich bevorechtigt)
CLOCK /OEI /AH /CS A7   A6    A5   A4   A0  GND
/OE   /DOE /O1 /O2 AWRT WRITE STOP READ OEO VCC

O1 :=      AH *  A4 * A5 * A6 * /O1
         +  AH *  O1 * O2 * A7
O2 :=      AH *  A4 * A5 * A6 * A7 * O1
         +  AH *  A4 + A5 * A6 * A7 * O2
         +  AH *  O1 * O2 * A7
DOE =      AH *  O1 * O2 * OEI * CS
OEO =    OEI * /AH
         + OEI * /O1
         + OEI * /O2
/AWRT  := /AH + /O1 + /O2 + /A7 +  A5 +  A4
/WRITE := /AH + /O1 + /O2 + /A7 +  A5 + /A4
/READ  := /AH + /O1 + /O2 + /A7 + /A5 +  A4
/STOP  := /AH + /O1 + /O2 + /A7 +  A6 +  A5
END
```

Dieses Listing gibt den Inhalt des Standard-PALs wieder.



Dazu ermittelt man die Länge des zu erweiternden Programms und linkt die Zeit- und Datums-Routinen an eine Adresse, die ein paar Bytes dahinter liegt. Im Debugger werden das Hauptprogramm und die Zeit-Routinen dann zusammengesetzt. An Adresse 100h schreibt man einen Sprung in die jetzt angefügten Zeitunterprogramme. Diese werden durch die drei bei 100h entfernten Bytes und einen Rücksprung ins Hauptprogramm abgeschlossen. Nun braucht man nur noch im Hauptprogramm zu suchen, wo Texte stehen, die später ausgegeben werden. Wird das HL-Register vor dem Sprung in die Zeit-Routinen mit diesen Adressen geladen, so schreiben die Uhrentreiber an die betreffenden Stellen die Uhrzeit und das Datum.

Wer seine Uhrentreiber selbst schreiben will, sollte folgendes beachten: Beim Auslesen der Uhr erscheint der Inhalt des angewählten Registers invertiert auf den unteren vier Bit des Datenbusses. BASIC-Programmierer sollten also vom gelesenen Wert 255 abziehen, bevor sie mit einem AND-Befehl die höherwertigen Datenbits ausmaskieren.

Normalerweise läuft die Uhr intern weiter, während man auf ihre Register zugreift. Nun kann es geschehen, daß ausgelesen wird, während die Zähler gerade weiterzählen. Dabei kann es zu Lesefehlern kommen. Wenn die Uhr beispielsweise gerade von 7:59:59 auf 8:00:00 Uhr springt, wechseln die Ziffern von rechts nach links auf Null, bevor der Stundenzähler auf 8 inkrementiert wird. Wird die Uhr genau in diesem Moment ausgelesen, erhält man möglicherweise die falsche Zeitangabe 7:00:00 Uhr. Dieser Fall kommt sehr selten vor, weniger als einmal auf 2000 Lesebefehle. Wenn man die Uhr also täglich fünfmal ausliest, dauert es im Mittel mindestens 1,5 Jahre, bis einmal ein falscher Wert auftritt. Deshalb wird diese Problematik von Programmierern oft einfach ignoriert.

Es gibt jedoch eine einfache Möglichkeit, den Zufall auszuschließen. Man liest die Uhr zweimal aus und vergleicht die Ergebnisse. Falls sie nicht übereinstimmen, wird ein drittes Mal gelesen, und das letzte Ergebnis wird an das rufende Programm übergeben.

Beim Schreiben tritt das Überlauf-Problem ebenso auf. Hier hilft uns jedoch die Hardware weiter: Alle Schreibzugriffe auf die Uhrenkarte werden mit A6 = 0 ausgeführt, auch schon der AWRT-Befehl. Dadurch wird das STOP-Signal an den Uhrenchip angelegt, um den Sekundentakt zu blockieren. Falls nun doch einmal ein Zählimpuls kommen sollte, kann er keinen Schaden mehr anrichten.

Zum Schluß noch ein Tip: Wenn alle Uhrenzugriffe ausgeführt sind, sollte man stets Register 15 selektieren, bevor man die Uhrenlogik deaktiviert. So geht man ganz sicher, daß beim Ein- und Ausschalten des Rechners nicht versehentlich einmal die Uhr verstellt werden kann. Im c't-Labor ist das allerdings auch ohne diese Vorbeugungsmaßnahme nicht passiert.

Untergeschoben

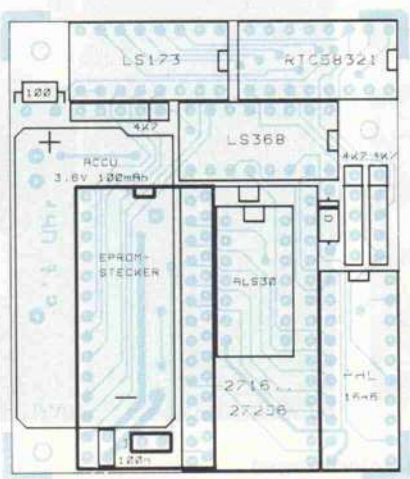
Am einfachsten ist der Einbau in den IBM PC oder einen seiner vielen Ableger. Man öffnet das Gehäuse, sucht sich eines der EPROMs auf Motherboard oder Interface-Karten aus und schiebt diesem die c't-Uhr unter die Beine. Nur das EPROM auf der Video-Interface-Karte dürfen Sie nicht nehmen, weil der Prozessor darauf keinen direkten Zugriff hat.

Man könnte nun Software schreiben, die die Hardware-Uhr ausliest und mit diesen Daten die Software-Uhr des Rechners stellt. Eleganter ist jedoch ein Device-Treiber, der über den c't-Software-Service erhältlich ist. Die Installation dieses

Stückliste

IC1	74ALS30/74LS30 (siehe Text)
IC2	EPROM (aus Rechner)
IC3	PAL16R6
IC4	74LS173
IC6	RTC58321
RSIL1...3	4k7 (4+1-Array)
R1	100 R
D	BSY21
C	100nF
Akku	3,6V/100mA (Mempack S)

EPROM-Fassung
28poliger DIL-Stecker
Platine



Vor dem Bestücken sollten Sie die Platzverhältnisse in Ihrem Computer prüfen und gegebenenfalls eine Variante mit Kabelanschluß wählen.

Programms ist sehr einfach: Der Name des Treibers wird in die Datei CONFIG.SYS eingetragen. Die c't-Uhr wird dadurch vollständig in das Betriebsprogramm eingebunden, egal, in welchem EPROM-Sockel sie steckt. Von nun an kann die Uhr über die DOS-Systemaufrufe von beliebigen Programmen gelesen und gestellt werden.

Im Gehäuse des Schneider CPC464 reicht der Platz zwischen Hauptplatine und Tastatur nicht aus, um die voll bestückte Uhrenkarte unterzubringen. Man muß deshalb den Akku über ein kurzes zweipoliges Kabel anschließen und ihn neben der Uhrenkarte in das Rechnergehäuse legen.

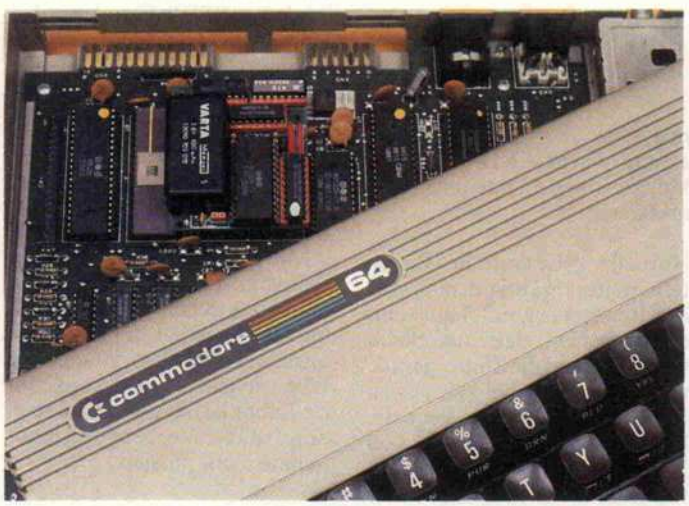
Etwas eng geht es auch beim Atari ST zu. Die Uhr paßt zwar in voller Pracht auf den hintersten EPROM-Sockel (neben dem ROM-Port), kollidiert dann aber mit dem oberen Abschirmblech. Schnelle Praktiker, die nicht gerade die Zimmerantenne von Radio oder Fernseher neben dem Rechner stehen haben, mögen vielleicht auf die Idee kommen, das Blech einfach wegzulassen. Wer im Einklang mit der Bundespost meint, daß das Ding doch einen Sinn hat, wird sich entweder mit dem beschriebenen Anschluß per Flachkabel behelfen oder ein Fenster in das Abschirmblech schneiden.

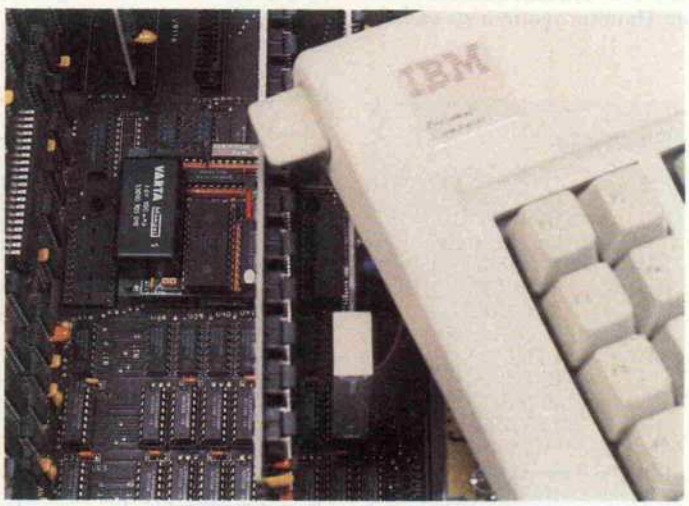
Der Einbau der Uhrenkarte in den C64 ist problemlos. In seinem dicken Bauch ist ausreichend Platz vorhanden. Dennoch bereitete die Ansteuerung der Uhr hier zunächst die größten Probleme. Das Oszilloskop förderte den Grund ans Licht: Im $\overline{CS}$-Signal des EPROMs waren Spikes zu finden, die auf dem Schirm so dicht wie die Bäume im Urwald standen. Diese haben ihre Ursache in dem Fehlen eines Memory-Request-Signals bei der 6502-CPU. Das CS-Signal kann also nicht synchronisiert werden; es wird direkt aus den Adressen gewonnen. Wegen der relativ langen Zugriffszeiten bei EPROMs und der niedrigen Taktfrequenz des C64 wirken sich die Störungen im Normalbetrieb nicht aus, die schnelle Steuerlogik der Uhr reagierte jedoch darauf und schaltete das EPROM unbeabsichtigt aus.

Abhilfe ließ sich dadurch schaffen, daß die Uhrenkarte nicht unter das Kernal-EPROM, sondern unter das BASIC-EPROM geschoben wurde. Wer einen C64 besitzt und die c't-Uhr einbauen möchte, sollte sich aber vor dem Aufbau der Uhrenkarte davon überzeugen, daß das BASIC-EPROM des Rechners gesockelt ist. An den Versuch, das EPROM auszulöten und nachträglich zu sockeln, sollte sich besser nur der erfahrene Bastler wagen. Eventuell können Sie Ihren Commodore-Händler damit beauftragen, diese Arbeit auszuführen. Verschiedene Firmen bieten diese Service-Leistung an, da aufgrund des Angebots an Hardware-Ergänzungen zum C64 ein entsprechender Bedarf entstanden ist.

Die Unterstützung der Uhrenkarte durch Software für den C64 wird unser Schwestermagazin INPUT 64 in seiner Ausgabe 4/86 übernehmen. Da INPUT 64 auf Diskette beziehungsweise Kassette erscheint, entfällt das lästige Abtippen von Listings. Assembler-Programmierer können sich den Uhrentreiber natürlich auf der Basis der abgedruckten Tabellen auch selbst schreiben.

Auch für die große Gruppe der Einplatinenrechner kann die c't-Uhr in der Regel problemlos nutzbar gemacht werden. Dadurch wird es endlich möglich, diesen häufig für Steuerungszwecke eingesetzten Rechnern





In vielen Rechnern läßt sich die c't-Uhr ganz problemlos unterbringen.

eine Aufgabe zu übertragen, bei der die Uhrzeit eine Rolle spielt.

Falls die Programme des Einplatinenrechners jedoch ausnahmslos im EPROM laufen, benötigt man eine modifizierte Version des PALs auf der Uhrenkarte. Denn bei der Standardausführung würde, wie schon erwähnt, bei jedem Opcode-Fetch der Uhrenzugriff wieder abgeschaltet. Das geänderte PAL jedoch, das unter der Bezeichnung 'Version E' über den Platinen-Service erhältlich ist, räumt einen kleinen Bereich ein, in welchem nicht abgeschaltet wird. Er beginnt 255 Bytes unterhalb der höchsten EPROM-Adresse und ist 128 Bytes lang. In diesem Adreßraum werden Zugriffe auf das EPROM von der Uhrenlogik ignoriert. Z80-Programmierer können dorthin zum Beispiel die abgedruckten Unterprogramme 'UHRLES' und 'UHRSCH' linken. Durch deren Aufruf können beliebige Programme aus dem EPROM problemlos die Uhr bedienen.

Übrigens ist es nicht erforderlich, die Uhr im Einplatinenrechner auch stellen zu können, denn sie verliert keine Daten, wenn man sie von einem Rechner zum anderen umsteckt. Wen eine Gangungenauigkeit von ein bis zwei Minuten im Jahr nicht stört, der kann die c't-Uhr einmal im PC stellen und sie danach im Einplatinenrechner vergessen. Denn die Schaltjahre werden automatisch korrigiert.

```
        TITLE   --> FORTIME <--   Zugriff auf die c't Uhr (auch) von FORTRAN
    .
;
;
;               $$$$$  $$$   $$$$          $$$$$  $$$  $    $  $$$$$
;               $     $   $  $   $  $        $     $   $$  $$  $
;               $$$   $   $  $$$$   $$$$$    $     $   $ $$ $  $$$
;               $     $   $  $ $$           $     $   $  $  $  $
;               $      $$$   $  $           $    $$$  $     $  $$$$$
;             ---------------------------------------------------------
;
;
;               UNTERPROGRAMME ZUR KOMFORTABLEN BENUTZUNG DER
;
;                               c't UNIVERSALUHR
;
;                           VON  F O R T R A N  PROGRAMMEN AUS
;
;
;       TIME    ERMITTELT DIE UHRZEIT (8 HOLLERITH ZEICHEN, Z.B. "15:03:28")
;       DATE    ERMITTELT DAS DATUM (12 ZEICHEN, Z.B. "Mo 12.Aug.82")
;
;       SETRTC  ZUM STELLEN DER UHR. PARAMETER: 7-STELLIGES INTEGER FELD,
;               MIT FOLGENDEN DATEN:  SEC/MIN/STD/WTAG/TAG/MON/JAHR.
;
;       BEISPIEL FUER DIE BENUTZUNG:
;               INTEGER TSTRIN(4), DSTRIN(6)
;               CALL TIME (TSTRIN)
;               CALL DATE (DSTRIN)
;               WRITE (3,100) TSTRIN, DSTRIN
;        100    FORMAT (5X,'Uhrzeit: ',4A2,  5X,'DATUM: ',6A2)
;
;
;
        NAME    ('TIMLIB')
        ENTRY   TIME    ;UHRZEIT IN ASCII ABLEGEN BEI: (HL)..(HL)+7
        ENTRY   DATE    ;DATUM IN ASCII ABLEGEN BEI: (HL)..(HL)+11
        ENTRY   SETRTC  ;ZEIT IN INTEGER VON (HL)..(HL)+14 LESEN & UHR SETZEN
;
        .Z80
        PAGE    65

GERMAN  EQU     0       ;=1 : WOCHENTAG & MONAT IN DEUTSCH, =0: ENGLISCH
;
EPROMS  EQU     0C000H          ;EPROM START ADR. (LOG.ADR.)
EPROML  EQU     4000H           ;EPROM GROESSE
UHR     EQU     EPROMS+EPROML-128 ;UHR ADRESSBEREICH: UHR .. UHR+127
;
NOSIG   EQU     UHR+70H ;SIGNALE FUER DIE UHR:
AWRT    EQU     UHR+40H
WRITE   EQU     UHR+50H
READ    EQU     UHR+60H
STOP    EQU     -40H    ;AUF AWRT,... ADDIEREN, UM STOP ZU SETZEN
;
;RTC50321 REGISTER NAMEN:
SEC1    EQU     0
SEC10   EQU     1
MIN1    EQU     2
MIN10   EQU     3
HOUR1   EQU     4
HOUR10  EQU     5
DOW     EQU     6
DAY1    EQU     7
DAY10   EQU     8
MON1    EQU     9
MON10   EQU     10
YEAR1   EQU     11
YEAR10  EQU     12
REFS    EQU     15
;
        SUBTTL  .       SETRTC          SETUP THE RTC
SETRTC: DS      0       ;UHR EINSTELLEN MIT BIN[REN DATEN.
;                        EIN (ODER MEHRERE) FALSCHER WERT WIRD IGNORIERT.
;       ENTRY   HL      ADR. EINES DATENBLOCKS MIT FOLGENDEN 16BIT WERTEN:
;                       SEC/MIN/HOUR/DAY-OF-WEEK/DAY-OF-MONTH/MONTH/YEAR
;       USES    AF
;
        PUSH    HL
        PUSH    DE
        PUSH    BC
        LD      BC,59*256+SEC1  ;REGISTER NUMMER, MAX. WERT
        CALL    SET2R           ;SETZE DEN SEKUNDEN ZAEHLER
;
        LD      BC,59*256+MIN1
        CALL    SET2R           ;MINUTEN
;
        LD      BC,23*256+HOUR1
        CALL    SETHOU          ;STUNDEN
;
        LD      BC,7*256+DOW
        CALL    SET1R           ;TAG DER WOCHE
;
        LD      BC,31*256+DAY1
        CALL    SET2R           ;TAG DES MONATS
;
        LD      BC,12*256+MON1
        CALL    SET2R           ;MONAT
;
        LD      BC,99*256+YEAR1
        CALL    SET2R           ;JAHR
;
        POP     BC
        POP     DE
        POP     HL
        RET
;-------
SET1R:  DS      0       ;EINEN 16BIT WERT LESEN UND IN DIE UHR SCHREIBEN
;       ENTRY   HL      ADRESSE DES DATUMS
;               B       MAX. WERT (BCD)
;               C       NUMMER DES ZU STELLENDEN UHRREGISTERS
;
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        CALL    MAKBCD
        RET     C               ;WERT NICHT ERLAUBT
        AND     0FH
        CALL    UHRSCH          ;O.K., TU'S
        RET
;-------
SET2R:  DS      0       ;16 BIT WERT LESEN, IN BCD WANDELN UND
;                        DAMIT ZWEI UHRREGISTER STELLEN.
;       ENTRY   HL      ADR. DER DATENW (WORD)
;               B       GROESSTER ERLAUBTER WERT (BCD)
;               C       NUMMER DES ERSTEN REGISTERS DER UHR
;
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        CALL    MAKBCD
        RET     C               ;WERT NICHT ERLAUBT
SET2RH: PUSH    AF
        AND     0FH             ;ZUERST DAS UNTERE NIBBLE
        CALL    UHRSCH
        INC     C
        POP     AF
        AND     0F0H
        RRCA
        RRCA
        RRCA
        RRCA
        CALL    UHRSCH
        RET
;-------
SETHOU: DS      0       ;WIE SET2R, SETZT JEDOCH DAS BIT FUER DEN 24STD-MODE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        INC     HL
        CALL    MAKBCD
        RET     C               ;FALSCHER WERT
        OR      80H             ;24H (STATT 12H) ZAEHLER
        JR      SET2RH
;**********************
;
        SUBTTL  .               T I M E         HOLE DIE ZEIT VON DER c't UHR
        PAGE
TIME:   DS      0       ;LEGE IN ASCII "STD:MIN:SEC" IM RAM AB
;       ENTRY   (HL)    ADRESSE DES ZIELFELDES (8 BYTES LANG)
;       EXIT            (HL)=(HL)+8
;       USES    ALL
;
        LD      A,HOUR10
        CALL    UHRLES          ;LIES STUNDEN 10ER
        AND     7               ;ENTFERNE DAS 24H-FLAG BIT
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      A,HOUR1
        CALL    UHRLES
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      (HL),':'
        INC     HL
        LD      A,MIN10
        CALL    UHRLES          ;MINUTEN 10ER
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      A,MIN1
        CALL    UHRLES
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      (HL),':'
        INC     HL
        LD      A,SEC10
        CALL    UHRLES          ;SEKUNDEN 10ER
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      A,SEC1
        CALL    UHRLES
        ADD     A,'0'
        LD      (HL),A
        RET
;**********************
        SUBTTL  .               D A T E         HOLE DAS DATUM VON DER c't UHR
        PAGE
DATE:   DS      0       ;LEGE DAS DATUM IN ASCII IM RAM AB
;       ENTRY   (HL)    ADRESSE DES ZIELFELDES (12 BYTES LANG)
;       EXIT            (HL)=(HL)+12
;       USES    ALL
;
        LD      A,DOW           ;WOCHENTAG (MONTAG = 1)
        CALL    UHRLES          ;LIES VON DER UHR
        ADD     A,A             ;DIE NAMEN DER WOCHENTAGE SIND ZWEI BYTES LANG
        EX      DE,HL           ;ZIELADRESSE IN DE
```

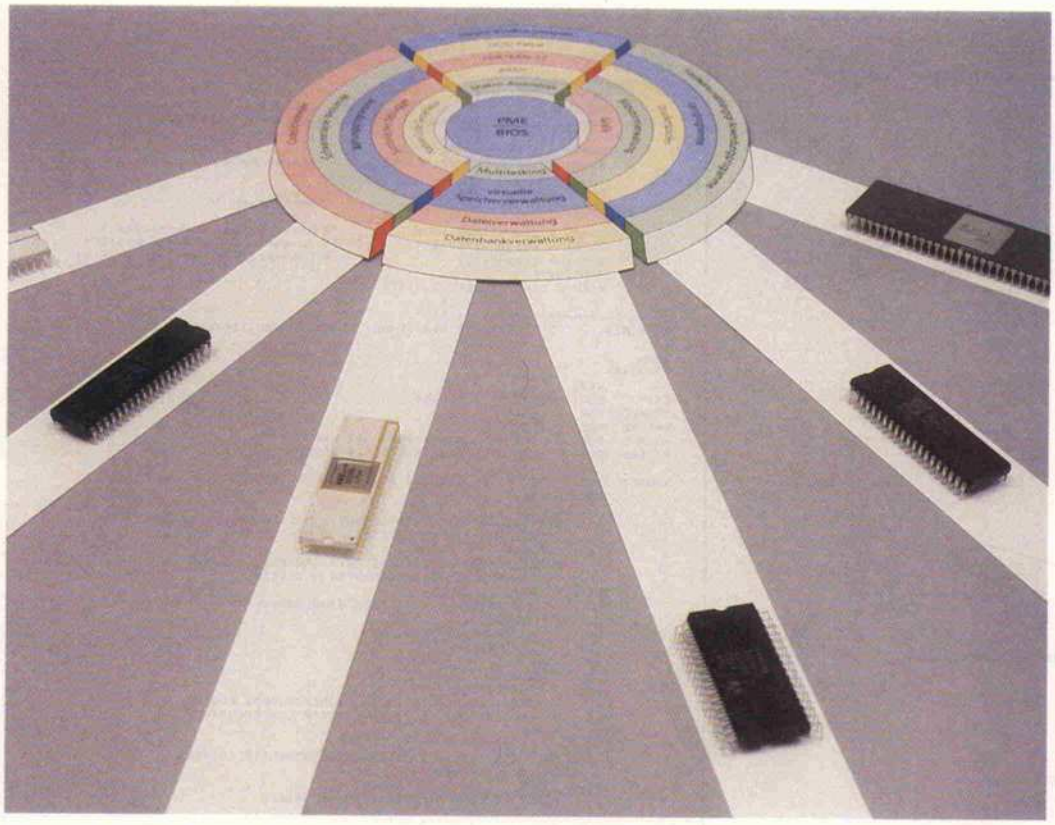
```
        LD      B,0
        LD      C,A
        LD      HL,DNAME-2
        ADD     HL,BC           ;HL = ADRESSE DES WO-TAG NAMENS
        LD      C,2             ;2 BUCHSTABEN
        LDIR
        EX      DE,HL
;
        LD      A,' '
        LD      (HL),A
        INC     HL
;
        LD      A,DAY10
        CALL    UHRLES          ;TAG (10ER)
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      A,DAY1
        CALL    UHRLES
        ADD     A,'0'
        LD      (HL),A
        INC     HL
;
        LD      (HL),'.'
        INC     HL
;
        LD      A,MON10         ;MONAT (10ER)
        CALL    UHRLES
        AND     A               ;WANDELE BINAER
        PUSH    AF              ;(ZERO FLAG)
        LD      A,MON1
        CALL    UHRLES
        LD      C,A
        POP     AF
        LD      A,0
        JR      Z,LESS10        ;MONATE 1..9
        LD      A,10
LESS10: ADD     A,C
        EX      DE,HL
        LD      HL,MNAME-3      ;ADR. DER MONATSNAMEN TABELLE
        LD      C,A
        ADD     A,C
        ADD     A,C             ;MULTIPLIZIERE MIT 3 (LAENGE DES MONATSNAMENS)
        LD      B,0
        LD      C,A
        ADD     HL,BC           ;HL = ADR. DES NAMENS
        LD      C,3
        LDIR                    ;KOPIERE DREI BUCHSTABEN
        EX      DE,HL
;
        LD      (HL),'.'
        INC     HL
;
        LD      A,YEAR10
        CALL    UHRLES          ;JAHR (10ER)
        ADD     A,'0'
        LD      (HL),A
        INC     HL
        LD      A,YEAR1
        CALL    UHRLES
        ADD     A,'0'
        LD      (HL),A
        RET
;
           IF GERMAN NE 0
DNAME:  DB      'MoDiMiDoFrSaSo'
MNAME:  DB      'JanFebMrAprMaiJunJulAugSepOktNovDez'
           ELSE
DNAME:  DB      'MoTuWeThFrSaSu'
MNAME:  DB      'JanFebMarAprMayJunJulAugSepOctNovDec'
           ENDIF
;
;***********************
        SUBTTL  .                       GEMEINSAME UNTERPROGRAMME
MAKBCD: DS      0       ;WANDELE BINAERES WORT IN BCD UM; MELDE, WENN ZU GROSS
;       ENTRY   DE      WERT
;               B       MAXIMAL ERLAUBTER WERT
;       EXIT    A       UMGEWANDELTE BCD ZAHL
;         ODER  C-FLAG  WENN B > DE
;       USES    F,B,D
;
        LD      A,D
        OR      A               ;MUSS NULL SEIN
        SCF
        RET     NZ
        LD      A,B             ;MAXIMUM
        CP      E               ;MUSS KLEINER ODER GLEICH B SEIN
        RET     C
;---                             DER WERT IST O.K., WANDELE UM:
        LD      A,E
        LD      D,0             ;NIMM DIE 10ER:
MAKDIV: SUB     10
        JR      C,MAKDIE        ;ZU VIEL ABGEZOGEN, ENDE DER SCHLEIFE
        INC     D
        JR      MAKDIV
;
MAKDIE: ADD     A,10            ;(WAR VORHER ZUVIEL ABGEZOGEN WORDEN)
        LD      B,A             ;BEHALTE DEN REST DER DIVISION IN B
        LD      A,D             ;10ER
        RLCA                    ;SHIEBE DIESE INS OBERE NIBBLE:
        RLCA
        RLCA
        RLCA
        ADD     A,B
        RET
;
;-----------------------
        SUBTTL  .       UNTERPROGRAMME FUER ZUGRIFF AUF DIE UHR
UHRSCH: DS      0       ;EIN REGISTER DER UHR BESCHREIBEN
;       ENTRY   C       NUMMER DES REGISTERS (0..15)
;               A       WERT (BEREICH: 0..9, ODER WENIGER,
;                                       z.B.MIN-ZEHNER:0..5)
;       USES    AF,B
;
        PUSH    HL
        PUSH    AF
        CALL    UHREIN          ;ZUGRIFF AUF UHR ERM\GLICHEN
        LD      A,C             ;REGISTERNUMMER
        LD      HL,AWRT+STOP    ;UHR ANHALTEN ZUM BESCHREIBEN
        ADD     A,L
        LD      L,A
        LD      A,(HL)          ;ADDR. WRT SIGNAL SETZEN
        LD      HL,NOSIG+STOP
        LD      A,C
        ADD     A,L
        LD      L,A
        LD      A,(HL)          ;WIEDER ZURUECKN., REG.# IST GESETZT
;
        LD      HL,WRITE+STOP
        POP     AF              ;DATEN
        AND     0FH
        ADD     A,L
        LD      L,A
        LD      A,(HL)          ;DATEN AUF UHR GEBEN
        NOP                     ;INSGESAMT ZUS WARTEN (Z80 BEI 8MHZ)
        LD      HL,NOSIG        ;(UHR LAUFT WIEDER AN)
        LD      A,(HL)
        CALL    UHRAUS          ;UHR AUS ADRESSRAUM ENTFERNEN & RETURN
        POP     HL
        RET
;-------
UHRLES::DS      0       ;EIN REGISTER DER UHR LESEN
;       ENTRY   A       NUMMER DES REGISTERS (0..15)
;       EXIT    A       WERT AUS REGISTER
;       USES    AF,B
;
        PUSH    HL
        CALL    UHREIN          ;UHRZUGRIFF EINSCHALTEN
        LD      B,A             ;REGISTERNUMMER SICHERN
        LD      HL,AWRT
        ADD     A,L
        LD      L,A
        LD      A,(HL)          ;ADDR. WRT SIGNAL SETZEN
        LD      HL,NOSIG
        LD      A,B
        ADD     A,L
        LD      L,A
        LD      A,(HL)          ;UND WIEDER ZURUECKNEHMEN

        LD      A,(READ)        ;DAS LESESIGNAL ANLEGEN
        NOP
        NOP                     ;(ER IST NICHT DER SCHNELLSTEN EINER..)
        LD      A,(READ)        ;DA SIND DIE DATEN
        CPL                     ;DER TREIBER INVERTIERT
        AND     00001111B
        CALL    UHRAUS          ;UHRZUGRIFF AUSSCHALTEN UND RETURN
        POP     HL
        RET
;-------
UHRW::  DS      0       ;AUF EINE NEUE SEKUNDE WARTEN
;               (UHRW KANN VOR ANDEREN UHRZUGRIFFEN AUFGERUFEN WERDEN
;                SIE VERHINDERT, DASS ZUFAELLIG DIE UHR GERADE INTERN
;                WEITERZAEHLT, WAEHREND MAN IHRE REGISTER LIEST.
;                DIES TRITT MIT EINER WAHRSCHEINLICHKEIT < 0,0005 AUF
;                UND HAT EINEN DATENFEHLER ZU FOLGE.
;                UHRW BRAUCHT NICHT VOR SCHREIBZUGRIFFEN AUFGERUFEN
;                WERDEN!                                          )
;
        XOR     A
        CALL    UHRLES          ;SEKUNDENZAEHLER HOLEN
        LD      E,A             ;SICHERN
UHRWL:  XOR     A
        CALL    UHRLES
        CP      E
        JR      Z,UHRWL         ;ZEAHLER IST NOCH NICHT UMGESCHLAGEN
;
        LD      B,120           ;GILT FUER Z80-CPU BEI MAX. 8MHZ
UHRWWL: DJNZ    UHRWWL          ;GIB DER UHR ZEIT, FERTIGZUZAEHLEN
        RET
;-------
UHREIN: DS      0       ;ZUGRIFF AUF UHR EINSCHALTEN
;       USES    B,HL
;
;       ---  DIE FOLGENDEN ZEILEN SCHALTEN DAS EPROM, IN DESSEN  ---
;       ---  SOCKEL DIE c't UHR STECKT, IN DEN ADDRESSRAUM DER   ---
;       ---  CPU EIN.                                           ---
;       ---  DIESE MUSSEN JE NACH RECHNER ANGEPASST WERDEN.      ---
;       ---  (HIER: SCHNEIDER CPC RECHNER)                      ---
        DI
        PUSH    BC
        LD      BC,0DF00H       ;EPROM EINBLENDEN:
        OUT     (C),C           ;ROM SELECT 0
        LD      BC,7F86H
        OUT     (C),C           ;BLENDE ROM EIN
        POP     BC
;       ---  (ENDE DES RECHNERSPEZIFISCHEN TEILS)               ---
;
        LD      HL,UHR+127      ;HOECHSTE ADR.
        LD      B,(HL)
        LD      B,(HL)
        LD      B,(HL)
        RET
;----
UHRAUS: DS      0       ;ZUGRIFF AUF UHR WIEDER AUSSCHALTEN
;       USES    HL
;
        PUSH    AF
        LD      A,(AWRT+REFS)   ;ZUR SICHERHEIT VOR STOERUNGEN BEIM
        LD      A,(NOSIG+REFS)  ;EIN- UND AUSSCHALTEN
;
        LD      A,(EPROMS)      ;UHRZUGRIFF DEAKTIVIEREN
        POP     AF
;
;       ---  DIE FOLGENDEN ZEILEN ENTFERNEN DAS EPROM, IN DESSEN  ---
;       ---  SOCKEL DIE c't UHR STECKT, AUS DEM ADDRESSRAUM DER   ---
;       ---  CPU.                                                 ---
;       ---  DIESE MUSSEN JE NACH RECHNER ANGEPASST WERDEN.       ---
        PUSH    BC              ;EPROM AUSBLENDEN:
        LD      BC,0DF07H
;               ODER BEI RAM ERWEITERUNG: 0DF06H,
;               BEIM CPC464 STEHT DER WERT IN DER SYSTEMVARIABEN BEI (0B1A8H)
        OUT     (C),C
        EXX                     ;JETZT GEHT'S IN DES SCHNEIDERS KATAKOMBEN ...
        OUT     (C),C           ;URSPRUNGLICHE RAM/ROM KONFIGUR. WIEDERHERST.
        EXX
        POP     BC
        EI
;       ---  (ENDE DES RECHNERSPEZIFISCHEN TEILS)               ---
        RET
;
        END
```

```
        PROGRAM TIMSET
        INTEGER TSTRIN(4), DSTRIN(6),FF,HT,SETT(7)
        DOUBLE PRECISION SNUM
        DATA FF/12/, HT/9/
C
        WRITE (3,100) FF,HT,HT
```

```
100     FORMAT (1X,A1/1X,A1,A1,' Einstellung der Hardware Uhr '/)
        GO TO 1100
1000    WRITE (3,101)
101     FORMAT (/' Setzen & auslesen / Ende? '
     *  '(S/E (<RETURN>=Kontrolle)): ')
        READ (3,20) SETT(1)
20      FORMAT (A1)
        IF (SETT(1).EQ.'e' .OR. SETT(1).EQ.'E') STOP
        IF (SETT(1).EQ.'s' .OR. SETT(1).EQ.'S') GO TO 2000
        GO TO 1100

2000    WRITE (3,105)
105     FORMAT (/' Bitte geben Sie die aktuelle Zeit ein.',
     *  ' (Ungueltige Eingaben werden ignoriert.'/
     *  ' Sie koennen eine Eingabe also mit 999 ueberspringen.)'//
     *  ' Sekunden: ')
        READ (3,10) SETT(1)
10      FORMAT (I2)
C
        WRITE (3,110)
110     FORMAT ('+MINUTEN:   ')
        READ (3,10) SETT(2)
C
        WRITE (3,120)
120     FORMAT ('+STUNDEN:   ')
        READ (3,10) SETT(3)
C
        WRITE (3,130)
130     FORMAT ('+WOCHENTAG: ')
        READ (3,10) SETT(4)
C
        WRITE (3,140)
140     FORMAT ('+MONATSTAG: ')
        READ (3,10) SETT(5)
C
        WRITE (3,150)
150     FORMAT ('+MONAT:     ')
        READ (3,10) SETT(6)
C
        WRITE (3,160)
160     FORMAT ('+JAHR(z.B.85): ')
        READ (3,10) SETT(7)
C                               ZEIT IN UHR UEBERTRAGEN
        CALL SETRTC (SETT)
C*******
        WRITE (3,200)
200     FORMAT (//' KONTROLLE: '/)

1100    CALL TIME (TSTRIN)
        CALL DATE (DSTRIN)
        WRITE (3,900) TSTRIN, DSTRIN
900     FORMAT (5X,'TIME:',4A2,  5X,'DATE:',6A2)
        GO TO 1000

        END
```

```
/*
   Die Routine DATE dient der Benutzung der c't Uhr unter der
   Programmiersprache C. Ihr geht ein kurzes Hauptprogramm voraus,
   das nach erfolgreichem Test entfernt wird.
   DATE ist mit dem C80 Compiler unter CP/M entstanden und
   in seiner Aufrufstruktur zumindest für diesen genormt. e.m.2/86
*/

$asm
        .request        clib    ;(Buechereien des C80 Systems)
        .request        crunlib
$endasm
$include fprintf.h
main()
{
  int year;
  char month,day,hour,min,sec;
  sec = date(&year,&month,&day,&hour,&min);

  printf ("\nDatum: %2d.%2d.%2d,",day,month,year);
  printf (" Uhrzeit: %2d:%2d:%2d!\n",hour,min,sec);
}


/* ------------------------------------------------------------------ */

#define EPROM_TOP  40960-1      /* höchste Adr. des EPROMs=9FFF */

int date(pyear,pmonth,pday,phour,pmin)
        /* Mit dieser Routine kann die c't Uhr direkt von C Programmen
           aus gelesen werden. In der Regel liegt das EPROM, in dessen
           Sockel sich die Uhr befindet, nicht staendig im Adressraum
           der CPU.
           Dann wird mit den Assembler-Unterprogrammen "UHREIN" und
           "UHRAUS" kurz vor dem Zugriff das EPROM eingeblendet und
           danach wieder abgeschaltet.
        */
        int     *pyear;
        char    *pmonth,*pday,*phour,*pmin;
{
        *pyear  = (lies_reg(11) + 10*lies_reg(12));
        *pmonth = (lies_reg(9) + 10*lies_reg(10));
        *pday   = (lies_reg(7) + 10*lies_reg(8));
        *phour  = (lies_reg(4) + 10*(lies_reg(5)&3));
        *pmin   = (lies_reg(2) + 10*lies_reg(3));
        return (lies_reg(0) + 10*lies_reg(1));
}
/* ------------------------- */
lies_reg (reg)  /* Lies ein Register der Uhr. Diese Routine wird dem
                   Hardware Zugriff (lies_hwr) zwischengeschaltet, um die
                   sehr seltenen Auslesefehler (1/2000, siehe Text) beim
                   RTC58321 gänzlich zu vermeiden.                    */
char reg;
{
   char zwi;
        zwi = lies_hwr(reg);
        if (lies_hwr(reg) == zwi)  return(zwi);
        return (lies_hwr(reg)); /* war Sek.Sprung, noch mal lesen */
}
/* --------------- Hardware-Zugriff: ----------------- */

#define UHR EPROM_TOP-127
#define AWRT  64        /* =0x40*/
#define READ  96        /* =0x60*/
#define NOSIG 112       /* =0x70*/
lies_hwr (regnum)       /* Hardwarezugriff: lies ein Register der Uhr */
char regnum;            /* Registernummer (0 für Sec .. 12 für Jahr) */
{
char dummy, wert, *ctuhr;
        ctuhr = UHR;
        UHREIN();       /* c't Uhr in den Adressraum der CPU schalten */
        dummy=ctuhr[127]; dummy=ctuhr[127]; dummy=ctuhr[127];

        dummy = ctuhr[regnum+AWRT];
        dummy = ctuhr[regnum+NOSIG];    /* Adressregister auswaehlen */

        dummy = ctuhr[READ];
        wert  = ctuhr[READ];            /* Wert lesen */

        dummy = ctuhr[AWRT+15];         /* Referenzreg. selektieren */
        dummy = ctuhr[NOSIG+15];        /* (dient nur zur Sicherheit) */
        dummy = ctuhr[-1];              /* Zugriffslogik deaktivieren */
        UHRAUS();
        return ((wert^0xff) & 0x0f);
}
/* -------------------------- */
UHREIN()        /* Schalte das Uhren- EPROM in den Adressraum der CPU */
{
$asm
;-------
        .Z80
DK      EQU     050863  ;thanks
SYSMODE EQU     0F3H
EPROMS  EQU     8000H
MMUD    EQU     0FEH    ;STEUERPORT DER MMU
EPROMP  EQU     1EH     ;PHYSIKALISCHES SEGMENT

;UHREIN::DS     0       ;ZUGRIFF AUF UHR EINSCHALTEN
;       USES    B,HL
;
;       ---  DIE FOLGENDEN ZEILEN SCHALTEN DAS EPROM, IN DESSEN  ---
;       ---  SOCKEL DIE c't UHR STECKT, IN DEN ADDRESSRAUM DER   ---
;       ---  CPU EIN.                                            ---
;       ---  DIESE MUSSEN JE NACH RECHNER ANGEPASST WERDEN.      ---
;       ---  (HIER: MMU MIT 256 SEGMENTEN A 4KBYTE:)            ---
        PUSH    AF
        LD      A,SYSMODE       ;EPROM EINBLENDEN:
        OUT     (MMUD),A
        LD      A,EPROMS/1000H
        LD      HL,0FFF0H
        OR      L
        LD      L,A
        LD      A,(HL)
        LD      (SEG1+1),A      ;ALTE SEGMENTNUMMER SICHERN
        LD      (HL),EPROMP     ;EINSCHALTEN DES EPROMS
        INC     HL
        LD      A,(HL)
        LD      (SEG2+1),A      ;ALTE SEGMENTNUMMER SICHERN
        LD      (HL),EPROMP+1
        POP     AF
;       ---  (ENDE DES RECHNERSPEZIFISCHEN TEILS)              ---
;
        RET
        .8080
$endasm
}
/* -------------------------- */
UHRAUS()                /* Uhrzugriff wieder abschalten */
{
$asm
        .Z80
;----
;UHRAUS::DS     0       ;ZUGRIFF AUF UHR WIEDER AUSSCHALTEN
;       USES    HL
;
;       ---  DIE FOLGENDEN ZEILEN ENTFERNEN DAS EPROM, IN DESSEN  ---
;       ---  SOCKEL DIE c't UHR STECKT, AUS DEM ADDRESSRAUM DER   ---
;       ---  CPU.                                                 ---
;       ---  DIESE MUSSEN JE NACH RECHNER ANGEPASST WERDEN.       ---
        PUSH    AF
        LD      A,SYSMODE       ;EPROM EINBLENDEN:
        OUT     (MMUD),A
        LD      A,EPROMS/1000H
        LD      HL,0FFF0H
        OR      L
        LD      L,A
SEG1:   LD      (HL),0          ;(HIER SETZT UHREIN DIE SEGMENTNR. EIN)
        INC     HL
sEG2:   LD      (HL),0          ;(2. SEGMENTNUMMER)
        POP     AF
;       ---  (ENDE DES RECHNERSPEZIFISCHEN TEILS)              ---
        RET
        .8080
$endasm
}
```

```
10 REM Das folgende Programm dient nur als ein Beispiel fuer den Zugriff
20 REM auf die c't Universaluhr von BASIC aus. Fuer IBM-PC Kompatible,
30 REM auf denen dieses Programm läuft, gibt es einen Einheitentreiber,
40 REM der die c't Uhr ins Betriebssystem integriert. Bei dieser wesentlich
50 REM eleganteren Loesung kann dann die Zeit einfach ueber die BASIC
60 REM Varablen DATE$ und TIME$ abgerufen werden.              e.m. 2/86
70 REM
80 DEF SEG = &HC9F0
90 REM  Diese Adresse gilt, falls die Uhr auf die Winchester/Controller
100 REM Karte gesetzt wird. Beim Einsatz auf der Mutterplatine muss man
110 REM beachten, dass DOS und BASIC u.U auch auf das EPROM zugreifen.
120 REM Die Uhr ist dann weg, noch bevor das BASIC Programm sie auslesen
130 REM konnte.
140 DIM REG(13)
150 FOR I=1 TO 13
160 R=I-1
170 GOSUB 1000
180 NEXT I
190 REG(5) = REG(5)-8
200 REM
210 PRINT "Datum:",REG(9);REG(8);".";REG(11);REG(10);".";REG(13);REG(12)
220 PRINT "Zeit:",REG(6);REG(5);":";REG(4);REG(3);":";REG(2);REG(1)
225 DEF SEG
230 STOP
1000 D=PEEK(&HFF)
1010 D=PEEK(&HFF)
1020 D=PEEK(&HFF)
1030 D=PEEK(&HC0+R)
1040 D=PEEK(&HF0+R)
1050 D=PEEK(&HE0)
1060 REG(I)=PEEK(&HE0)
1070 D=PEEK(0)
1080 REG(I)=(REG(I) XOR 255) AND 15
1090 RETURN
```



CPU-Gleichrichtung

Das UCSD-p-System

Heinz Ebert

Die Entrüstung der Turbo-Pascal-Freaks ist vorprogrammiert: 'Oh, nein! Jetzt beschäftigt sich c't auch noch mit UCSD-Pascal!' Die in dieser abwertenden Bemerkung implizierte Verknüpfung von Pascal und p-System stammt aus den Kindertagen dieses Betriebssystems, als es tatsächlich noch UCSD-Pascal hieß und nur ein Pascal-Dialekt mit einer in Pascal geschriebenen Programmierumgebung war. Heute wird das 'p' im Namen nicht mehr von Pascal abgeleitet, sondern von der programmtechnischen Verwirklichung durch eine Pseudo-Maschine. Das UCSD-p-System stellt ein hochwertiges Betriebssystem dar, das viele nützliche Werkzeuge für die Software-Entwicklung bereitstellt, nach denen sich mancher Turbo-Programmierer die Finger lecken würde. Ein weltweiter Marktanteil von zwölf Prozent hat es 1984 an den dritten Platz aller verkauften Mikrocomputer-Betriebssysteme gebracht.

Die Aufgaben von Computer-Betriebssystemen haben sich im Verlauf der jetzt rund dreißigjährigen Geschichte der EDV gewandelt. Ursprünglich waren sie dafür entwickelt worden, die teuren Rechneranlagen immer optimal auszulasten, um möglichst wenig kostbare Rechenzeit zu verlieren. Durch manuelle Bedienung war das nicht zu erreichen. Als dann die Zahl der angeschlossenen Ein-/Ausgabeeinheiten zunahm, wurde auch deren Verwaltung mehr und mehr von den Betriebssystemen übernommen. Die damaligen Betriebsprogramme besaßen jedoch komplizierte Steuersprachen und konnten daher nur von Spezialisten bedient werden.

Die Personal Computer, die ja gerade Laien ständigen Zugriff auf Computerleistungen ermöglichen sollen, spannten für die Betriebssysteme eine vollkommen neue Problemdimension auf. Die kryptischen Geheimsprachen, deren Kenntnis die Systemspezialisten auszeichnete, mußten in 'menschenwürdige' Anwenderschnittstellen umgewandelt werden. Der Weg von CP/M über MS-DOS zu den modernen mausgesteuerten Piktogramm-Betriebssystemen zeigt deutlich die Wegmarken dieser langen, qualvollen Strecke.

Vergangenheit

Die vorbildliche Anwenderoberfläche des p-Systems setzte schon sehr früh ein Zeichen und ist durch seine Entwicklungsgeschichte zu erklären. Die Unzufriedenheit der Informatik-Anfänger mit den Unzulänglichkeiten der vorhandenen Rechenanlage regte an der University of California, San Diego (UCSD) ein Projekt an, das 1974 unter der Leitung von Professor K. Bowles gestartet wurde. Man versuchte, die hinderlichen Wartezeiten des überlasteten Timesharing-Systems durch die Verwendung vieler kleiner Rechner zu vermeiden und den Computerneulingen preisgünstig interaktiven Zugriff zu ermöglichen. Die Beschränkung dieser Systeme (PDP 11) auf nur 64 KByte Internspeicher und zwei Diskettenlaufwerke wurde durch den Einsatz des symbolischen p-Code-Prozessors und des dafür bereits vorhandenen Pascal-Compilers aufgefangen, die an der ETH Zürich bei Professor N. Wirth entwickelt worden waren.

Als im Jahre 1978 angekündigt wurde, daß das System für 8080/Z80-Mikrocomputer verfügbar sei, löste das einen Nachfrage-Boom aus. Entscheidenden Anteil am Siegeszug hatte auch die Implementation auf Rechnern mit einem 6502-Mikroprozessor. Ein Jahr darauf entzog sich die Universität der kaufmännischen Belastung, indem sie das System an die neugegründete Firma Softech Microsystems in San Diego weitergab, die dann als Lizenzgeber für Wartung und Fortentwicklung zuständig war. Später spaltete sich die Firma Volition Systems ab, die einen Modula-2-Compiler und einen verbesserten Editor für das p-System anbot. Beide Firmen sind mittlerweile in die Firma Pecan, New York, aufgegangen.

Den Übergang von der 8- zur 16-Bit-Technologie konnte das p-System problemlos überstehen, da es von Anfang an wort-

Gegenwart und Zukunft

orientiert ausgelegt war. Als Reaktion auf die ständig wachsende Adressierungsfähigkeit der Mikroprozessoren wurde die nutzbare Speicherkapazität auf 128 KByte angehoben. Inzwischen ist aber auch diese Grenze gefallen. Seit einiger Zeit ist sogar Multiuser- und Multitasking-Betrieb möglich (siehe Pinnacle Test, c't 1/86).

Jetzt ist die Schlacht um den 32-Bit-Mikrocomputer-Markt voll entbrannt. Der Anfangsoffensive von NS32032, NS32332 und MC68020 stellen sich Transputer und iapX386 entgegen. Das p-System nimmt die Herausforderung an, mit einer vollkommen neuentwickelten 32-Bit-p-Maschine wird es wahrscheinlich als Doppelagent für alle Gegner tätig werden. Die alte 8/16-Bit-Maschinerie soll aber auch im neuen System zur Verfügung stehen, um die bisherigen Programme auch weiterhin nutzen zu können.

Die historische Entwicklung und die seiner Architektur zugrundeliegenden Strategien erklären die Eigenschaften des p-Systems. Die drei wichtigsten Entwurfskriterien waren: Komfort bei der Bedienung, Maschinenunabhängigkeit von Betriebssystem und Anwenderprogrammen sowie kompakter Programmcode.

Lernhilfe

Damit ein Mikrocomputer-Betriebssystem als benutzerfreundlich gelten kann, sind zwei gegensätzliche Ansprüche miteinander zu vereinbaren: Die Anwendung soll für einen Anfänger leicht erlernbar sein und mit möglichst wenig Anweisungen auskommen. Einem erfahrenen Programmierer aber müssen alle Facetten des Systems zugänglich sein. Hier kollidiert natürlich die Forderung nach eingeschränkten Freiheitsgraden als Lernvorteil mit dem Wunsch nach schrankenloser Freiheit.

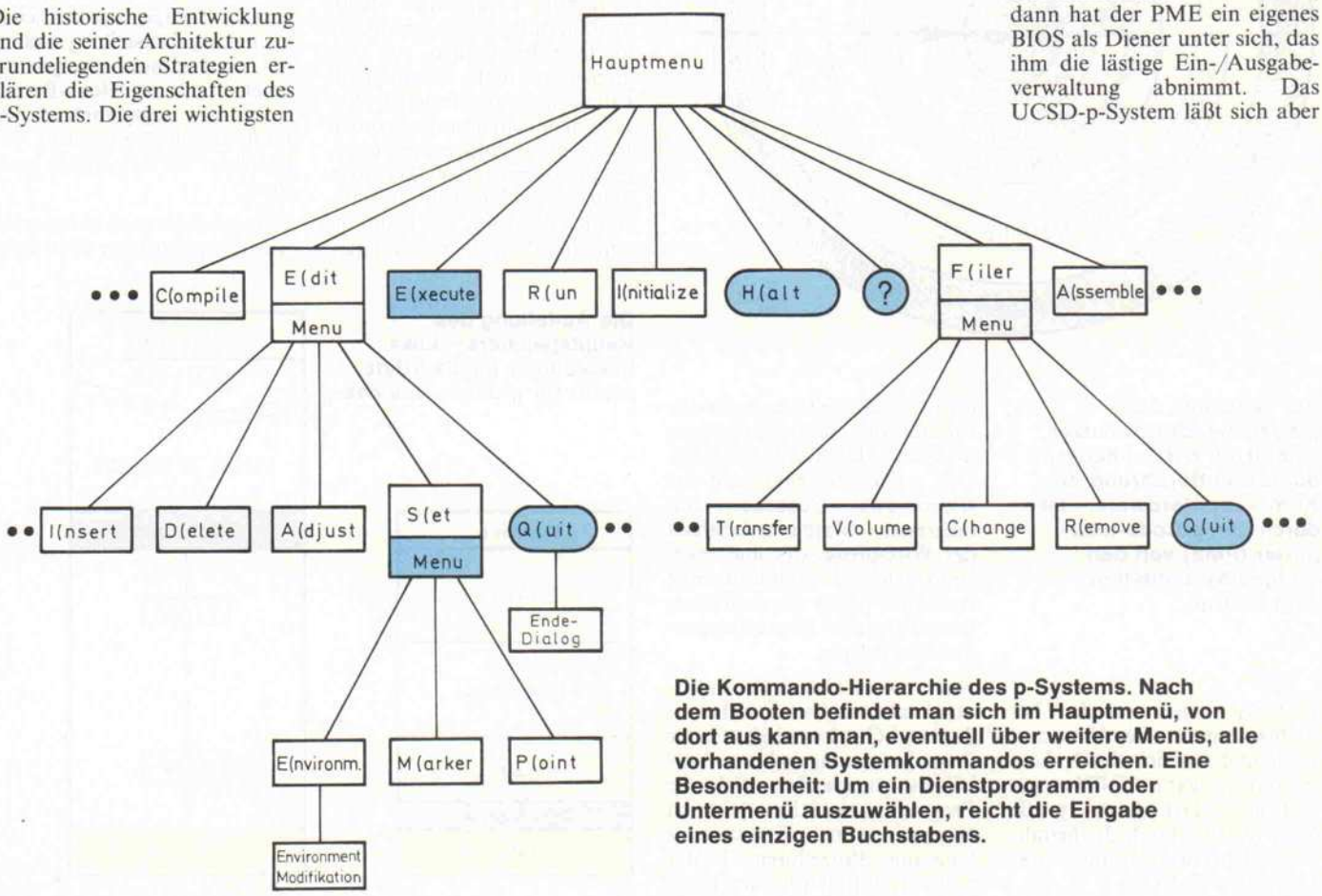
Die im p-System gegebene menügesteuerte und in Ebenen gegliederte Kommandoeingabe stellt einen vernünftigen Kompromiß dar. Die oberste Zeile auf dem Bildschirm enthält das jeweils aktive Menü, das den Benutzer ständig über den augenblicklichen Zustand des Systems und die bestehenden Wahlmöglichkeiten informiert. Diese Art der Organisation reduziert die Berührung eines Programmieranfängers mit dem Betriebssystem zunächst auf nur wenige Kommandos. Als 'Zauberlehrling' hat er die Hilfeleistung der Menüzeile ständig zur Verfügung. Später, mit wachsender Erfahrung, wird er diese Information ganz ignorieren.

Portables Kuckucksei

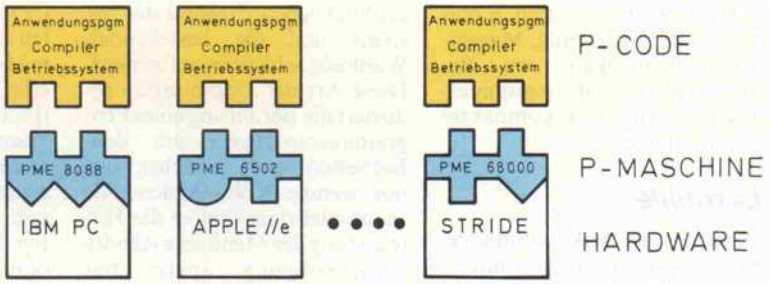
Aufmerksame Beobachter des Mikrocomputer-Softwaremarktes haben sicher schon bemerkt, daß das UCSD-p-System meist kurze Zeit nach dem Erscheinen eines neuen Computers als erstes zusätzliches Betriebssystem lieferbar ist. Dies wird durch das zweite wichtige Designkriterium ermöglicht, die Portabilität. Der Einsatz einer Pseudo-Maschine, die als kleines Programm im Maschinencode des Wirtsprozessors verwirklicht ist, gewährleistet ein sehr hohes Maß an Maschinenunabhängigkeit. Für jede neue CPU, auf der das p-System laufen soll, muß lediglich ein neues, p-Maschinen-Emulator (PME) oder auch p-Code-Interpreter (PCI) genanntes Assemblerprogramm entwickelt werden. Alle anderen Teile des Betriebssystems arbeiten bereits mit diesem Pseudo-Prozessor und lassen sich durch diesen Trick, ohne Änderungen vorzunehmen, von Maschine zu Maschine transportieren.

Eine zweite und häufigere Ursache von Inkompatibilitäten bildet das I/O-System. Neben unterschiedlichen Geräteschnittstellen und frei wählbaren Portstrukturen führen auch verschiedene Möglichkeiten der Ein-/Ausgabesteuerung dazu, daß kein Rechner einem anderen gleicht. Seit der Einführung von CP/M für 8080/Z80-Mikrocomputer gibt es aber ein probates Mittel, diese Schwierigkeiten elegant zu umgehen – das BIOS. Diese Ansammlung von Unterprogrammen vermittelt zwischen hardware-spezifischen Eigenheiten eines Rechners und den allgemeinen Anforderungen des Betriebssystems.

Wenn das p-System alleine über die Rechnerressourcen herrscht, dann hat der PME ein eigenes BIOS als Diener unter sich, das ihm die lästige Ein-/Ausgabeverwaltung abnimmt. Das UCSD-p-System läßt sich aber



Die Kommando-Hierarchie des p-Systems. Nach dem Booten befindet man sich im Hauptmenü, von dort aus kann man, eventuell über weitere Menüs, alle vorhandenen Systemkommandos erreichen. Eine Besonderheit: Um ein Dienstprogramm oder Untermenü auszuwählen, reicht die Eingabe eines einzigen Buchstabens.



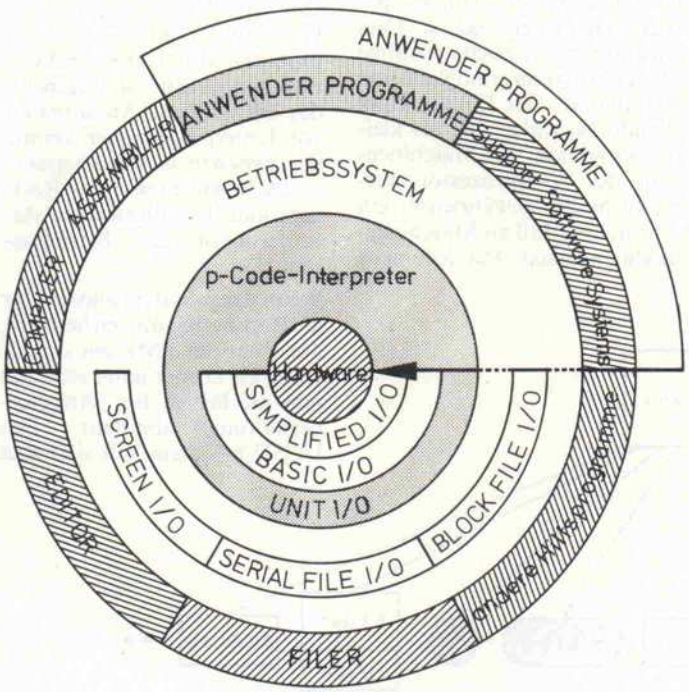
Die P(seudo)-Maschine macht's möglich: Für unterschiedliche Rechner braucht man nur den jeweils passenden PME. Die p-Code-Teile des Systems bleiben immer dieselben.

auch wie ein Kuckucksei in ein fremdes Betriebssystem legen und mißbraucht dann schmarotzend dessen Ein-/Ausgabesystem für seine Zwecke. Machte das p-System anfangs nur Anleihen beim CP/M-BIOS, sitzt es mittlerweile huckepack auch auf den Betriebssystemen vieler Minicomputer und hat sich durch diese Hintertür sogar in UNIX eingeschlichen. Der letzte große Coup gelang durch die heimliche Unterwanderung von MS-DOS/PC-DOS.

Gesetzliches Hindernis

Bei der Konzeption eines Hochsprachen-Systems für die frühen Mini- und Mikrocomputer stellte die geringe Internspeicherkapazität ein großes Problem dar, das jedoch durch den Einsatz des sehr kompakten p-Codes gelöst werden konnte. Die p-Code-Instruktionen ersetzen jeweils Sequenzen von mehreren Maschinenbefehlen des Wirtsprozessors und ergeben somit Lademodule, die nur etwa ein Fünftel von dem Platz entsprechender Maschinenprogramme belegen.

Nach einem Gesetz der Informatik aber stehen Speicherbedarf und Geschwindigkeit einer Prozedur in umgekehrtem Verhältnis zueinander. Schnelle Programme benötigen daher im allgemeinen mehr Speicher als langsame Prozeduren. Leider bildet die Emulation einer Pseudomaschine keine Ausnahme von dieser Regel. Der reduzierte Bedarf an Disketten- und Internspeicherplatz bringt unweigerlich einen Geschwindigkeitsverlust mit sich, p-Code-Programme beschäftigen die Wirts-CPU fünf- bis zwanzigmal so lange wie funktionsgleiche Assemblerprogramme. Mittlerweile hat man allerdings auch schon Rechner mit in Hardware gegossener p-Maschine und Varianten davon entwickelt, die diesen Nachteil nicht besitzen.



Die Schichten des p-Systems. Der Benutzer kommt nur mit der äußeren Schicht in Berührung. Der Kern – die Hardware – ist durch den p-Code-Interpreter (PME) von den übrigen Systemteilen abgeschirmt.

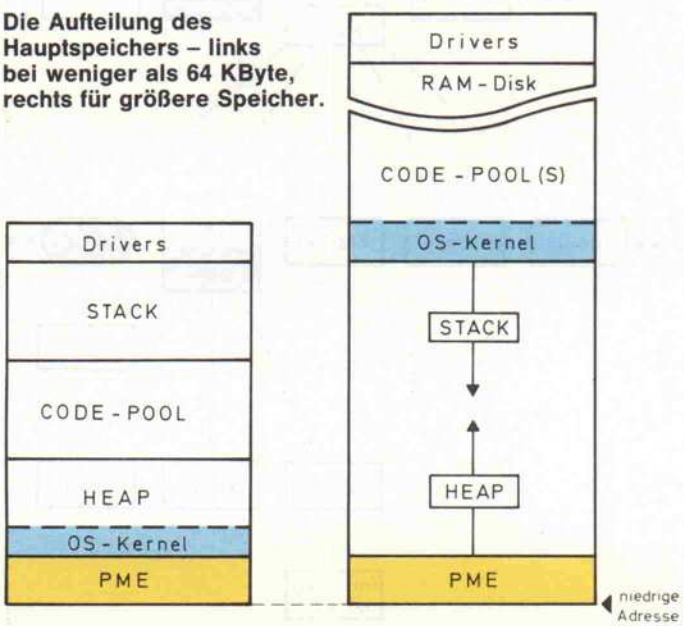
Strikte Trennung

Beim Bau moderner Software werden Modularisierung und Schichtung als Strukturierungswerkzeuge eingesetzt. Beide Verfahren benutzen genau definierte Schnittstellen zur Abgrenzung und Kontaktaufnahme der verschiedenen Programmeinheiten untereinander. In einem in Schichten gegliederten Programmsystem fließen die Daten ausschließlich in senkrechter Richtung, das heißt nur von einer Schicht zur anderen, und auch der Kontrolltransfer erfolgt von oben nach unten oder umgekehrt. Die Zerlegung in Module dient der horizontalen Abgrenzung von Prozeduren, also innerhalb einer Ebene in der Programmhierarchie.

Auch das p-System ist wie eine Zwiebel schichtenweise aufgebaut (siehe nebenstehende Grafik). Die innerste, maschinenabhängige Schicht bildet der p-Maschinen-Emulator. Er enthält auch das BIOS und schottet die außenliegenden Ebenen von der Hardware ab. An diesen Kern schmiegt sich das eigentliche Betriebssystem, das den Zugriff zum Emulator kontrolliert und bereits ganz in p-Code implementiert ist. Die Prozeduren des Betriebssystems werden dann von den Dienst- und Anwenderprogrammen in der äußeren Schale benutzt, die als eigentliche p-System-Oberfläche dem Benutzer zugänglich sind.

Dieses Schichtenmodell auf die Einteilung des Internspeichers zu übertragen, ist allerdings nicht sinnvoll. Hier geht man danach vor, wo man mit festem Speicherbedarf rechnen kann und wo nicht. Eine feste Größe haben zum Beispiel der PME und die Hardware-Treiber (Drivers). Abhängig von der jeweiligen Aufgabenstellung und damit variabel sind dagegen der Stack- und der Heap-Bereich sowie der sogenannte Code-Pool, die Betriebssystem und Emulator für ihre Arbeit benötigen.

Der von oben nach unten orientierte Stapelspeicher dient außer



Die Aufteilung des Hauptspeichers – links bei weniger als 64 KByte, rechts für größere Speicher.

der Buchhaltung über die Unterprogrammaufrufe auch der Auswertung von arithmetischen und logischen Ausdrücken und stellt Speicherplatz für statische Programmvariablen bereit. Auf dem Heap, der sich von unten nach oben ausdehnt und meistens vom Betriebssystem benutzt wird, werden dynamische Datenstrukturen und bei Programmstarts unverschiebliche Codesegmente abgelegt. Stack und Heap belegen zusammen maximal 64 KByte und wachsen hauptsächlich in LIFO-Manier aufeinander zu, dabei klemmen sie in Rechnern mit weniger als 128 KByte Memory noch den Code-Pool zwischen ihren Fronten ein.

Im Code-Pool stehen die verschieblichen Code-Segmente eines Programms, die zur Laufzeit dynamisch ein- und ausgelagert werden können (Overlay-Technik). Computer mit mehr als 128 KByte Speicher können mehrere jeweils 64 KByte große Code-Pools besitzen, in denen ab Revision IV.2.1 auch dynamische Datensegmente angelegt werden können.

Kriminell

Was sich die p-Maschine dem p-Code gegenüber erlaubt, erfüllt den Tatbestand der Hochstapelei, denn sie tut so, als ob sie eine echte CPU sei. In Wirklichkeit ist sie aber nur ein symbolischer Prozessor, dessen Hardware im Speicher des Wirtsrechners durch ein Programm nachgeahmt wird. Die Simulation einer Maschine nennt man Emulation. Die Elemente der p-Maschine sind:

1. Ein Satz spezieller Register
2. Der Stack
3. Der Befehlssatz
4. Die p-Maschinen-Emulator-Routine

Die neun Spezialregister des Registersatzes haben alle eine festgelegte Aufgabe. Sechs davon dienen als Zeiger auf umfangreiche Datenstrukturen, die für die Verwaltung von segmentierten Programmen zuständig sind. Für die Basisfunktion der p-Maschine von Bedeutung sind das IPC-Register, das als Instruktionszähler eingesetzt wird, und das SP-Register, welches den Stackpointer darstellt. Das neunte, IORESULT genannte Register signalisiert den Status von abgeschlossenen Ein-/Ausgabe-Aktionen.

Der Stack ist ein integraler Bestandteil der p-Maschine, die keinerlei frei verfügbare Arbeitsregister besitzt und auch noch ohne Akkumulator auskommen muß. Bei genauerem Hinsehen entpuppt sich der Stack jedoch als pulsierender allgemeiner Registersatz, der quasi-unbeschränkt immer so viele Register zur Verfügung stellt, wie man braucht. Sein oberstes Element, durch das SP-Register markiert, übernimmt dabei die Funktion eines Akkumulators. Die kleinste Arbeitseinheit von Stack und p-Maschine ist ein 16-Bit-Wort (zwei Bytes), ein Erbe der Erstimplementation auf einer 16-Bit-Maschine.

Der Befehlssatz der p-Maschine umfaßt 217 p-Code-Instruktionen. Da der Operationscode maximal ein Byte belegt, stehen noch 39 Opcodes für Erweiterungen zur Verfügung. Dem Operationscode können maximal drei Parameter folgen, sie enthalten Informationen über Größe und Anzahl der zu bearbeitenden Operanden. Ein Parameter kann auch selbst als Operand benutzt werden.

Die p-Maschine verwendet folgende Datentypen als Operanden:

1. Adressen (inklusive NIL)
2. Zeiger (Offsets, Byte- und Packed-Array-Zeiger)
3. Worte (typfreier Wert)
4. Werte (BOOLEAN, INTEGER, REAL)
5. Felder (Mengen, Wortblöcke, Aktivierungs-Records).

Als symbolisches Äquivalent zum Microcode-Dekoder und -Sequencer einer realen Mikroprozessor-Architektur ist der Emulator anzusehen. Je nach Betrachtungsweise kann man ihn auch als p-Code-Interpreter bezeichnen. Das bereits erwähnte IPC-Register zeigt auf die nächste auszuführende p-Code-Instruktion in einem Code-Segment eines Programms. Der Interpreter, ein Maschinenprogramm der Wirts-CPU, lädt diese Instruktion und benutzt den Operationscode als Index in eine zentrale Sprungleiste, entlang der die Adressen der Simulationsroutinen wie Perlen an einer Schnur aufgereiht sind. Die Unterprogramme am Sprungziel, die den so dekodierten Befehl stellvertretend für die imaginäre p-Maschine ausführen, sind

Mnemonic	Opcode Dez	Hex	Funktion
ABI	224	E0	Absolute Value Integer
ABR	227	E3	Absolute Value Real
ADI	162	A2	Add Integer
ADJ	199	C7	Adjust Set
ADR	192	C0	Add Real
ASTR	235	EB	Assign String
BNOT	159	9F	Boolean Not
BPT	158	9E	Breakpoint
CAP	171	AB	Copy Array Parameter
CFP	151	97	Call Formal Procedure
CGP	145	91	Call Global Procedure
CHK	203	CB	Check Subrange Bounds
CIP	146	92	Call Intermediate Procedure
CLP	144	90	Call Local Procedure
CSP	172	AC	Copy String Parameter
CSTR	236	EC	Check String Index
CXG	148	94	Call External Global
CXI	149	95	Call External Intermediate
CXL	147	93	Call External Local
DECI	238	EE	Decrement Integer
DIF	221	DD	Set Difference
DUP1	226	E2	Duplicate One Word
DUPR	198	C6	Duplicate Real
DVI	141	8D	Divide Integer
DVR	195	C3	Divide Real
EFJ	210	D2	Equal False Jump
EQBYT	185	B9	Equal Byte Array
EQPWR	182	B6	Equal Set
EQREAL	205	CD	Equal Real
EQSTR	232	E8	Equal String
EQUI	176	B0	Equal Integer
FJP	212	D4	False Jump
FJPL	213	D5	False Jump Long
FLT	204	CC	Float
GEBYT	187	BB	Greater Than or Equal Byte Array
GEPWR	184	B8	Greater Than or Equal Set
GEQI	179	B3	Greater Than or Equal Integer
GEREAL	207	CF	Greater Than or Equal Real
GESTR	234	EA	Greater Than or Equal String
GEUSW	181	B5	Greater Than or Equal Unsigned
INC	231	E7	Increment
INCI	237	ED	Increment Integer
IND	230	E6	Index
INN	218	DA	Set Membership
INT	220	DC	Set Intersection
IXA	215	D7	Index Array
IXP	216	D8	Index Packed Array
LAE	155	9B	Load Extended Address
LAND	161	A1	Logical And
LAO	134	86	Load Global Address
LCO	130	82	Load Constant Offset
LDA	136	88	Load Intermediate Address
LDB	167	A7	Load Byte
LDC	131	83	Load Constant
LDCB	128	80	Load Constant Byte

55 der 217 Befehle der p-Maschine. Mit jeweils bis zu drei Parametern werden die Operanden eines Befehls beschrieben, zum Teil sind die Parameter selbst die Operanden. Statt allgemeiner Register benutzt die p-Maschine für alle Berechnungen den Stack, dort werden auch die Ergebnisse abgelegt.
Der Befehl 'Load Constant Byte' zum Beispiel lädt ein als Parameter übergebenes Byte auf den Stack, die zu einem 16-Bit-Wort fehlenden acht (höherwertigen) Bits werden dabei automatisch auf '0' gesetzt. In der Assemblersprache der p-Maschine schreibt man dafür: LDCB ⟨UB⟩ (0 ≤ UB ≤ 255)

Microcode und Sequencer in einem. Als Beispiel ist hier ein Ausschnitt aus einem Unterprogramm abgedruckt, das REAL-Ausdrücke in umgekehrt polnischer Notation berechnet und alle die Funktionselemente besitzt, die auch der p-Code-Interpreter für seine Arbeit benutzt. Durch die ähnliche Struktur und Aufgabe wird auch eine ähnliche Arbeitsweise impliziert, so daß dieses kleine Programm sehr gut die grundlegende Funktion des Emulators zeigen kann.

Sexy Bytes

Leidgeprüfte Ehefrauen haben es schon immer vermutet. Jetzt steht es fest: Je mehr Bytes, desto sexyier der Computer. Nüchtern betrachtet handelt es sich bei diesem Byte-Sex jedoch nur um die Art und Weise, wie die CPU, auf der das p-System jeweils gastiert, 16-Bit-Worte aus byteorientierten Speichern liest. Einige Mikroprozessoren erwarten das höherwertige Byte eines Wortes vor dem niederwertigen, andere genau umgekehrt. Die p-Maschine selber nimmt standardmäßig immer den Byte-Sex der Wirts-CPU an. Probleme treten nur dann auf, wenn übersetzte Programme auf Maschinen mit unterschiedlichem 'Geschlecht' laufen sollen. Ein Flag in den Datenstrukturen als primäres Geschlechtsmerkmal eines Code-Segments signalisiert dessen Byte-Sex. Vor der Ausführung von andersgeschlechtlichen Code-Segmenten, die von einem Rechner mit entgegengesetztem Byte-Sex abstammen, wird deren Wortstruktur durch einmaliges Vertauschen von Bytes durch das Betriebssystem angepaßt.

Echte Schwierigkeiten gibt es nur mit den Daten-Files von Anwenderprogrammen, da deren Struktur dem Betriebssystem vollkommen unbekannt ist. Die Handbücher empfehlen ziemlich lakonisch, diese Programme eben so schlau zu machen, daß sie mit diesem Problem alleine fertig werden.

Neben der Simulation der p-Maschine muß der Emulator auch noch einige andere zeitkritische Operationen ausführen und mit dem BIOS zusammenarbeiten. Für diesen Zweck enthält der Emulator ein 'Run Time Support Package' (RSP) genanntes Unterprogrammpaket. Der Teil des RSP, der für die Ein-/Ausgabe zuständig ist, heißt RSP/IO und ist ein hardwareunabhängiges Interface zum gerätespezifischen BIOS.

```
.......

CONST ILL=0;  ADD=-1; SUB=-2; NEG=-3; MUL=-4;
      TEI=-5; POW=-6; FN1=-7; ....... FN28=-34;

TYPE KODTYP = -127..128;
     VALARR = ARRAY[1..MAXVALANZ] OF REAL;
     KODARR = PACKED ARRAY[1..MAXKODANZ] OF KODTYP;

.......

FUNCTION INTERPRETER(VAR Values     : VALARR;
                         KodeAnzahl : INTEGER;
                     VAR KodeArray  : KODARR) : REAL;

CONST MAXSTKLEN=30;

VAR Register           : KODTYP;
    Counter,TopofStack : INTEGER;
    Accu               : REAL;
    Stack              : ARRAY[0..MAXSTKLEN] OF REAL;

  PROCEDURE PUSH;
  BEGIN
    TopofStack:=TopofStack+1;
    IF (TopofStack > MAXSTKLEN) THEN RUNERR(-1);
    Stack[TopofStack]:=Accu
  END;

  FUNCTION POP : REAL;
  BEGIN
    POP:=Stack[TopofStack];
    TopofStack:=TopofStack-1;
    IF (TopofStack < 0) THEN RUNERR(-2)
  END;                                              } 1

BEGIN
  TopofStack:=0;  Counter:=0;  Accu:=MR;
  WHILE (Counter < KodeAnzahl) DO BEGIN
    Counter:=Counter+1;
    Register:=KodeArray[Counter];
    CASE Register OF
      ILL     :  RUNERR(-3);
      ADD     :  Accu:=POP+Accu;
      SUB     :  Accu:=POP-Accu;
      NEG     :  Accu:=-Accu;
      MUL     :  Accu:=POP*Accu;
      TEI     :  Accu:=POP/Accu;
      POW     :  Accu:=POWER(POP,Accu);
      FN1     :  Accu:=ABS(Accu);
      .
      .
      .
      FN28    :  Accu:=TANH(Accu);
      OTHERWISE  IF (Register > 0) THEN BEGIN
                   PUSH;
                   Accu:=Values[Register]
                 END
                 ELSE RUNERR(3)                     } 2
    END
  END;
  INTERPRETER:=Accu
END;
```

3 → CASE Register OF

4 ← RUNERR(3)

Dieser Ausschnitt aus einem Pascal-Unterprogramm, das umgekehrt polnisch notierte REAL-Ausdrücke berechnet, enthält alle grundlegenden Elemente des PME:

1 – definiert die im weiteren benötigten Stack-Operationen.

2 – beinhaltet die 'Microcodierung' der implementierten Befehle.

3 – ist der Dekoder und Sequencer für die abzuarbeitenden Befehle und

4 – der Error-Trap für ungültige 'Opcodes'.

Die Unit-I/O-Routinen des RSP werden mit den p-Code-Instruktionen CXG oder SCXG1 angesprungen und sind daher auch von Pascal-Programmen aus erreichbar, die sich am Betriebssystem vorbeischummeln möchten. Das RSP transformiert die p-Systemspezifischen Ein-/Ausgabe-Aufrufe in die entsprechenden BIOS-Aufrufe. Das BIOS schließlich fängt all die Variationen auf, die bei den I/O-Schnittstellen möglich sind. Dies geschieht durch Aufruf der einzelnen gerätespezifischen Bedienungsroutinen einer als 'Simplified BIOS' (SBIOS) bezeichneten Unterprogrammsammlung.

Eine einfache Ein-/Ausgabe-Anforderung durchläuft also mehrere Stationen, bis sie endlich am gewünschten Zielgerät ankommt. Im Kern des p-Systems sind bereits drei Stationen zu durchlaufen. Die Ein-/Ausgabe-Geräte arbeiten im allgemeinen weitaus langsamer als die CPU, daher fällt dieser Umweg normalerweise nicht ins Gewicht. Das eigentliche Betriebssystem, der Mittler zwischen Anwenderoberfläche und dem PME, fügt auch noch zwei Ebenen zur I/O-Hierarchie hinzu.

Vertikales Gewerbe

Das Betriebssystem arbeitet wie ein Speiseaufzug in einem Restaurant. Es transportiert Ein-/Ausgabe-Aufträge nach unten zum p-Maschinen-Kern. Nach Abschluß der Aktionen erfolgt die Rücksendung und -meldung nach oben zu dem anfordernden Anwendungs- oder Dienstprogramm.

Im Aufzugsschacht werden jedoch heimlich noch einige Veränderungen vorgenommen. Zuerst manipulieren die Laufzeitroutinen der Programmiersprachen die zu übertragenden Daten durch Formatumwandlungen, zum Beispiel durch Transformation von ASCII nach binär. Auf der nächsttieferen Ebene werden die I/O-Anfragen in blockorientierte und zeichenorientierte Datenströme sortiert und die I/O-Parameter für das Run Time Support Package zurechtgestutzt. Auf blockstrukturierten Geräten erlaubt das UCSD-p-System vier verschiedene Dateitypen: DIREC-

TORY-, DATA-, TEXT- und CODE-Files, die aus 512 Byte großen Blöcken bestehen. Der Platz auf den Datenträgern wird allerdings nicht in Blöcken, sondern in Dateien verwaltet. Dadurch entstehen bei der Rückspeicherung veränderter Files Lücken, die von Zeit zu Zeit eliminiert werden müssen, durch einen vom Anwender getriggerten Kompressionsbefehl namens Crunch. Die zeichenorientierte Ein-/Ausgabe wird noch in seriellen Datentransfer und Bildschirm-Kommunikation unterteilt, damit das Betriebssystem spezielle Manipulationen mit den Daten von und zur Systemkonsole vornehmen kann.

Außer dieser Funktion als Durchreiche für I/O-Aufträge übernimmt das Betriebssystem auch die Verantwortung für Heap- und Code-Pools. Nach dem Booten liegt der KERNEL genannte Betriebssystemkern bei kleinem Internspeicher selber auf dem Heap, ansonsten setzt er sich im externen Code-Pool fest. Dieser residente Kern enthält die Verwaltungsroutinen für die Code-Pools, die Fehlerbehandlung und den Lader für Programmsegmente.

Der freie Heap-Bereich wird vom Heap-Manager markiert, bevor die Kontrolle an ein Anwendungsprogramm übergeben wird, das dann dynamische Datenstrukturen auf dem Heap generieren kann. Nach der Terminierung des Programms wird der Heap bis zur Markierung wieder freigegeben. Alle Programme sind unter UCSD-p in Segmente aus maximal 255 Unterprogrammen gegliedert. Diese Segmente können zur Laufzeit des Programms dynamisch in die Code-Pools ein- und ausgelagert werden. Die Lücken von eliminierten Segmenten werden durch Nachschieben der restlichen automatisch geschlossen. Die Datensegmente, eine relativ neue Errungenschaft des p-Systems werden im Code-Pool wie Code-Segmente behandelt; sie verbleiben aber bis zum Ableben des Programms im System.

Das Betriebssystem stellt auch die Hilfsmittel für Multitasking bereit. So wie ein Kind eine Seifenblase nach der anderen erzeugt, kann ein aktives Programm andere, Tasks genannte, Prozesse absetzen. Die parallele Verarbeitung wird durch Task-Semaphore kontrolliert und von der p-Maschine mittels der zwei p-Maschinen-Befehle SIGNAL und WAIT gesteuert. Das Betriebssystem übernimmt nur die Aktivierung und Beendigung der Prozesse.

Für Änderungen am Betriebssystem wird das Konfigurierungsprogramm SETUP verwendet. Damit läßt sich das Verhalten von Bildschirm und Tastatur einstellen, also ob ein Memory-Mapped-Video-Interface oder ein normales RS-232-Terminal als Systemkonsole verwendet wird. Sehr wichtig ist die Möglichkeit, die interne Speicherbelegung des p-Systems ändern zu können. Auch ob eine Uhr vorhanden ist, ob Druck-Spooling erlaubt sein soll und wie viele zusätzliche serielle Schnittstellen zur Verfügung stehen, wird mit SETUP angepaßt. Dauerhafte Modifikationen des Betriebssystems, die auch das Abschalten des Computers überstehen sollen, müssen in der Datei SYSTEM.MISCINFO abgespeichert werden. Sonst betreffen die Änderungen nur die im Internspeicher residenten Datenstrukturen, was für Testzwecke sicher anzuraten ist.

Komponentenkleber

Wie das Bild mit der Kommando-Hierarchie zeigt, sind die p-System-Dienstprogramme und ihre Leistungen als Baum gegliedert. Die oberste Kommandoebene, die Wurzel des Baumes, wird vom Hauptmenü gebildet. Es leimt die selbständigen Module zur nutzbaren Systemoberfläche zusammen. Einige Dienstprogramme besitzen wiederum eigene Untermenüs, während andere in Dialogform mit dem Benutzer kommunizieren. Ein Wechsel von einer Option zur anderen innerhalb derselben Kommandoebene gelingt nur über das jeweilige Vorgängermenü. Der Benutzer kann also nicht wie ein Eichhörnchen von einem Ast (Option) zum anderen springen, sondern muß wie eine Schnecke den Umweg über die Astgabel (Menü) wählen. Als Wiedergutmachung dient ein Klammeraf-Zeichen hinter den Anfangsbuchstaben der Optionsnamen als optischer Hinweis dafür, daß schon die Eingabe des ersten Zeichens zur Anwahl genügt. Zum Beispiel startet ein im Hauptmenü eingegebenes 'C' den Compiler, ein 'E' den Editor.

Mit dem Set-Kommando im Hauptmenü lassen sich einige nützliche Änderungen am Erscheinungsbild der Systemoberfläche anbringen. Neben der Möglichkeit, Datum und Uhrzeit zu setzen, kann vor allen Dingen der augenblicklich aktive System-Compiler, -Assembler oder -Editor ausgetauscht werden, so daß auch andere Übersetzer mit den Systemkommandos erreichbar sind.

Von besonderer Bedeutung für den Anwendungskomfort ist, daß der Name des System-Workfiles, den die Software-Entwicklungsprogramme für den Quelltext oder als Lademodul verwenden, vorgegeben ist. Dadurch erspart man sich die lästige Eingabe des Programmnamens in der unvermeidlichen Entwicklungsschleife: Ändern, Übersetzen, Test, erneute Änderung und so weiter. Ohne die Möglichkeit, unterschiedliche Dateien als Workfile zu deklarieren, wäre das Ganze aber nur halb soviel wert. Auch hier hilft das Set-Kommando.

Das p-System erlaubt, die Standard-Ein-/Ausgabe-Einheiten auf Diskdateien umzulenken, zum Beispiel kann man mit dem Monitor-Kommando die Eingabe von Befehlen in einer Textdatei protokollieren. Eine durch Kommando-Monitoring erzeugte Befehlssequenz wird dann durch die Umleitung der Systemeingabe auf die entsprechende Datei geweckt. Auf diese Weise kann sich ein 'fauler' Programmierer eine Bibliothek mit Superkommandos anlegen.

Die Endbenutzer eines Anwendungsprogramms – beispielsweise einer Buchhaltungs-Software – sollen und wollen normalerweise vom Betriebssystem nichts merken. Zu diesem Zweck besitzt das UCSD-p-System eine Tarnkappe, unter der es seine eigene Oberfläche verbergen kann. Dieses für Laien nützliche Versteckspiel des Betriebssystems wird folgendermaßen erzielt: Ein in der Datei SYSTEM.STARTUP gespeichertes Programm unterdrückt das Hauptmenü nach dem Boot-Vorgang, allerdings nur solange es aktiv ist. Das Codefile SYSTEM.MENU dagegen wird immer anstelle der Systemkommandozeile verwendet. Auf diese Art und Weise lassen sich schlüsselfertige Anwendersysteme auf Diskette erzeugen. Diese Turnkey-Disketten brauchen dann auch nur die für eine einwandfreie Funktion eines Programms notwendigen Teile des Betriebssystems zu enthalten.

Gut bestückt

Das p-System stellt im Grunde genommen eine homogene Entwicklungsumgebung für Programme dar, und so ist es nicht verwunderlich, daß alle Komponenten auf ein optimales Zusammenspiel abgestimmt sind. Im Werkzeugkasten des Betriebssystems findet der Software-Tüftler eine Menge nützlicher Dienstprogramme, mit denen sich gut große Programme basteln lassen.

Das wichtigste Arbeitsmittel für einen Programmierer ist der Editor, der die Eingabe und Modifizierung von Quelltexten gut unterstützt. Die wenigsten Textverarbeitungsprogramme können mit so wenigen Kommandos ein solches Maximum an Benutzerfreundlichkeit vermitteln wie der im p-System eingebaute Editor. Nur vier Cursor-Tasten und einige Kommandozeichen genügen für effiziente Textbearbeitung. Befehle für Einfügen, zeilen- oder zeichenweises Löschen, Ersetzen und Blocktransport werden durch Kommandos für einfaches und ersetzendes Suchen von Text-Strings, seitenweises Blättern und Springen im Text ergänzt. Die bei Orthopäden als Patientenlieferant so beliebte 'Geierkralle' (Control- plus sonstige Taste) kommt bei diesem Editor nur für den Abschluß der Eingabe (Control-C) zum Einsatz. Wer auf seiner Tastatur noch eine Sondertaste frei hat, kann diesen Schönheitsfehler durch Umkonfigurieren mit SETUP beheben. Der Editor wird durch PRINT, den Druckformatierer des p-Systems, zum vollständigen Textsystem ergänzt.

Für die Verwaltung von Dateien auf Datenträgern (Volumes) enthält das UCSD-p-System einen File-Manager – kurz Filer genannt. Die wichtigsten Aufgaben des Filers neben der Ausgabe der Inhaltsverzeichnisse (Directories) von Datenträgern sind Kopieren, Umbenennen und Löschen von einzelnen Dateien, Dateigruppen oder ganzen Disketten. Dateigruppen werden mit Jokerzeichen – auch als Wild Cards bekannt – angesprochen, vorausgesetzt, ihre

Mitglieder enthalten gleichlautende eindeutige Teil-Strings im Namen.

Durch das unglückliche File-Handling des p-Systems entstehen im Laufe der Zeit Lücken auf den Disketten, die ab und zu eliminiert werden müssen. Der Filer kann solche 'zerklüfteten' Datenträger komprimieren. Andere Filer-Optionen suchen defekte Datenblöcke auf einem Datenträger oder listen die augenblicklich aktiven Volumes und Geräte auf.

Das Directory eines Volumes kann maximal 77 Dateien enthalten, was der Übersichtlichkeit durchaus förderlich ist, bei Festplatten aber eher stört. Um auch solche großen Datenträger effizient zu nutzen, können im Hauptinhaltsverzeichnis sogenannte Subsidiary Volumes für jeweils 77 weitere Dateien angelegt werden. Diese gelten dann als eigenständige Datenträger und müssen explizit eröffnet und geschlossen werden.

Editor und Filer stellen die für die Eingabe und Verwaltung der Software notwendigen Systemfunktionen zur Verfügung. Die Compiler und Assembler, die zweite wichtige Gruppe der UCSD-p-Dienstprogramme, bilden die Basis der Programmentwicklung.

Konzertierte Sprachaktion

Die Dienstprogramme der Betriebssystem-Oberfläche einschließlich der Compiler sind alle schon in UCSD-Pascal geschrieben. Der Pascal-Compiler ist daher der ausgereifteste Übersetzer unter den drei verfügbaren Hochsprachen. Vollintegrierte String-Bearbeitung, BCD-Arithmetik, Multitasking und das UNIT-Konzept sind Sprachelemente, die über das normale Standard-Pascal hinausgehen. Compileroptionen und bedingte Übersetzung können über spezielle Kommentare eingestellt werden.

Die Sprachpalette wird durch einen FORTRAN-77-Compiler erweitert, der strukturierte Programmierung unterstützt und ebenfalls Zeichen- und String-Variable erlaubt. IF-THEN-ELSE-Statements gibt es auch für den BASIC-Compiler. Eine erwähnenswerte Besonderheit des UCSD-BASIC sind die virtuellen Felder, die bis zu 32767 Elemente enthalten können und auf externe Speichermedien abgebildet werden. Die Verwaltung übernehmen die Laufzeitroutinen, im Programm werden sie wie normale Variablen verwendet.

Zeitkritische Aufgaben und andere maschinennahe Probleme werden selbst im Zeitalter der Sprache 'C' immer noch bevorzugt in der Assembler-Sprache des Zielprozessors programmiert. Ein BIOS in Spaghetti-Code wird wohl noch einige Zeit die Systementwicklung beherrschen. Unter dem p-System ist folgerichtig auch jeweils ein Assembler für die aktuelle Wirtsmaschine vorgesehen. Ein Wermutstropfen trübt jedoch die Freude: Die p-System-Assembler verwenden eine eigene Syntax, die nicht mit der 'natürlichen' Syntax der Mikroprozessorhersteller korrespondiert. Zum Ausgleich dafür handelt es sich durchwegs um Macro-Assembler, die auch über eine Schnittstelle zur p-Maschine verfügen. Fertig übersetzte Assembler-Routinen können mit dem Link-Befehl an Hauptprogramme gebunden werden. Aber damit ist das Füllhorn des p-Systems noch nicht leer, denn es gibt auch ein Crossassembler-Paket für alle gängigen Mikroprozessoren außer für die NS32000-Chips. (Warum?)

Der Hauptvorteil der UCSD-Sprachfamilie: Die Unterprogramme der drei Hochsprachen und auch in Assembler kodierte Routinen können gemeinsam in einem Hauptprogramm verwendet werden. Einige nicht miteinander zu vereinbarende Spracheigenschaften führen zu Einschränkungen, die hauptsächlich durch unterschiedliche Implementation der Datentypen hervorgerufen wird. Standard-Pascal hat – im Gegensatz zu FORTRAN und BASIC – durch sein flexibles Typen- und Variablen-Konzept Schwierigkeiten mit Modulbibliotheken, die vorübersetzte Prozeduren und Funktionen enthalten. Die Compilerkonstrukteure des p-Systems hatten sich schon sehr früh eine Verbesserung für modulare Programmierung einfallen lassen. Das Basiskonzept dieser Verbesserung findet man heute eingebaut in Modula-2 wieder, das eigentlich mehr ein erweitertes Pascal als eine neue Sprache ist.

Wenn die Natur vor 3,5 Gigajahren nicht die selektive Zellwand erfunden hätte, würden Sie, verehrter Leser, jetzt nicht,

Einheitliche Privatsphäre

gemütlich in der Wanne sitzend, diesen Artikel lesen, sondern das Bad schon nach kurzer Zeit als gestaltloser Proteinklumpen (Amöbe) verlassen. Das biologische Prinzip: 'So wenig Außenwelt wie möglich, und nur so viel Außenwelt, wie unbedingt notwendig' *) wurde in die UNITs des UCSD-p-Systems übernommen. Die funktionsmäßig zusammengehörenden Prozeduren und Datenstrukturen einer Unit werden in einen öffentlichen und einen privaten Bereich unterteilt. Der INTERFACE-Teil kennzeichnet die exportierbaren Ressourcen einer Unit, während die im IMPLEMENTATION-Abschnitt definierten Arbeitsmittel absoluter Geheimhaltung und Exportbeschränkung unterliegen. Ein Programm, das Werkzeuge aus solchen Einheiten importieren will, signalisiert dies dem Compiler mit dem USES-Statement, das alle gewünschten Units aufzählt. Mit den Angaben aus den INTERFACE-Abschnitten der verlangten Einheiten kann der Compiler das gesamte Programm übersetzen, ohne den Quelltext der Unit-Unterprogramme zu benötigen.

Bei der Erstellung werden die Units zweckmäßigerweise mit dem LIBRARY-Hilfsprogramm in Bibliotheken zusammengefaßt, von wo aus sie beim Start eines Hauptprogramms hinzugeladen werden. Der erzielte Nettoeffekt: Eine fertige Unit kann als Black Box aufgefaßt werden. Sie gibt von ihrem Innenleben nur das Format der Ein- und Ausgabedaten und den Namen der für die Bearbeitung notwendigen Unterprogramme preis. Der interne Verarbeitungsalgorithmus versteckt sich hinter der IMPLEMENTATION-Barriere wie der Beamtenapparat eines Ministeriums hinter den Schranken der Pförtnerloge.

Die System-Library SYSTEM.PASCAL enthält wichtige Units von Betriebssystem und Dienstprogrammen. Die Einheit SCREENOPS zum Beispiel besteht aus Routinen für die Bildschirmansteuerung, während die FILER-Units einem Programm die Leistungen des Filer-Dienstprogrammes zur Verfügung stellen. Die SPOOLER-Einheit erlaubt den Ausdruck eines Textes als parallele Hintergrundaufgabe und die ERROR-HANDLER-Unit ermöglicht es einem Programm, seine eigenen Laufzeitfehler abzufangen. Außer solchen relativ harmlosen Einheiten gibt es allerdings auch Units, die schwerwiegende Eingriffe in das Verhalten des Emulators und die Verwaltung des Internspeichers auslösen.

Große Programme lassen sich unter dem p-System regelrecht in kleine Internspeicher falten, da Unterprogramme als Segmentroutinen definierbar sind, die zur Laufzeit je nach Bedarf ein- und ausgelagert werden. Mit den Units kann man genauso verfahren. Außerdem erlaubt ein Chaining genannter Vorgang, andere Hauptprogramme zu starten. Dabei können den neuen Programmen Daten und Parameter übergeben werden.

Durch die verschiedenen Spielarten der Overlay-Technik kann ein Software-Entwickler seine Programme zu einem logisch geordneten Systemgebirge auftürmen. Als Folge dieser flexiblen Strukturierungsmöglichkeiten schleppen die p-Code-Lademodule umfangreiche Datenstrukturen mit sich, die erst in der Startphase vom Betriebssystem endgültig aufgebaut werden. Das QUICKSTART-Hilfsprogramm kann diese Datenstrukturen eines Programms in einem separaten Arbeitsgang erzeugen und abspeichern. Auf diese Art und Weise verkürzt es entscheidend die Ladezeiten segmentierter Programme.

Aufrüstung

Für gehobene Ansprüche läßt sich das p-System auch durch mächtige Werkzeuge für die Programmentwicklung ausbauen. Hinter einer 'Advanced Development Tool Kit' genannten Programmsammlung verbergen sich in der Hauptsache ein symbolischer Debugger, ein Native Code Generator, ein p-Code-Disassembler und ein Patch-Programm für Disk-

*) Hoimar von Ditfurth, Der Geist fiel nicht vom Himmel, dtv, München, 1980

Files. Der Native Code Generator (NCG) ist besonders erwähnenswert, er wandelt markierte Unterprogramme innerhalb eines p-Code-Lademoduls in den Maschinencode des Wirtsprozessors um. So können zeitkritische Programmteile ohne Assembler optimiert werden. Eine Einschränkung bedeutet, daß der NCG bei Unterprogrammen mit Real-Arithmetik versagt. Für jede Real-Operation wird ein Rücksprung in den Emulator eingebaut, wodurch sich die Verarbeitungszeit paradoxerweise sogar um bis zu fünfzig Prozent erhöht.

Eine weitere Spezialität ist beispielsweise das Programmpaket namens POPTYSER, das Optimierungen auf p-Code-Ebene erlaubt, aber nichts für einen schmalen Geldbeutel ist (3200 DM). Oder die KSAM-Bibliothek, deren Units den Aufbau von leistungsfähigen Datenbanken ermöglichen. Die Einheiten des INSIGHT Window-Design-Paketes übernehmen die Verwaltung von verschieblichen Fenstern und Pull-Down-Menüs in Programmen. Der normale Systemeditor kann durch den EDVANCE ersetzt werden, der unter anderem beliebig lange Texte und mehrere Textdateien gleichzeitig bearbeiten kann. Das Angebot an Anwendungsprogrammen beinhaltet Datenbanken mit und ohne Pascal-Schnittstelle, Tabellenkalkulation sowie umfangreiche kaufmännische Software für Lager- und Kundenstammverwaltung, Buchhaltung und Mahnwesen.

Abschlußzeugnis

Das große Reservoir an Software-Entwicklungswerkzeugen ist sicher ein schlagkräftiges Argument für das p-System. Weitaus bedeutsamer ist aber die Tatsache, daß das p-System infolge seiner portablen Struktur auf über 150 Rechnertypen und für fast alle gängigen Mikroprozessoren implementiert ist. Kein anderes Betriebssystem kann sich auf eine so breite Basis an Computern stützen. Mit dem Betriebssystem ist aber auch die darauf entwickelte Software von Rechner zu Rechner übertragbar. Die Möglichkeit, die verschiedenen Hochsprachen nicht nur einzeln verwenden, sondern in einem Programm auch mehrsprachig arbeiten zu können, stellt ein weiteres Plus für das UCSD-p-System dar.

Von den Einwänden gegen das p-System ist der systemimmanente Verlust an CPU-Zeit der schwerwiegendste. Die verminderte Effizienz wird inzwischen durch die immer schnelleren Mikroprozessoren und ihre hochspracheorientierten Befehlssätze gemildert, aber ein gewisser Reibungsverlust durch die Emulation wird bleiben. Dennoch steht das UCSD-Pascal gar nicht so schlecht da. Die häufig zitierte kürzere Übersetzungszeit von Turbo-Pascal wird insbesondere bei größeren Programmen durch die Verwendung vorkompilierter Modulbibliotheken voll aufgefangen. Bis heute verfügt Turbo-Pascal außer auf Quelltextebene über kein Sprachelement, das modulare Programmierung unterstützt.

Die Handbücher zeichnen sich mehr durch Quantität (über 4500 Seiten laut Werbung) als durch Qualität aus. Sie nutzen zu können, setzt schon eine gewisse Vertrautheit mit dem System und seiner Architektur voraus. Für Anfänger gibt es einige empfehlenswerte Bücher, die didaktisch hervorragend gleichzeitig in das p-System und in die Programmiersprache Pascal einführen.

Um ein immer wieder leidiges Thema nicht zu vergessen: Mit dem UCSD-p-System gut beraten ist, wer zukunftsorientiert und bequem große Anwendungsprogramme entwickeln will und die Kosten steuerlich absetzen kann. Der Paketpreis für Betriebssystem plus Pascalcompiler beträgt 1640 DM und ist damit höher als der Preis für einen bekannten, unter deutschem Firmennamen vertriebenen Homecomputer mit semiprofessioneller Ausbaufähigkeit. Die einfache Nutzungslizenz für UCSD-p kostet 890 DM. Wenn man sie einzeln nachkauft, muß man für Fortran- oder Pascalcompiler satte 990 DM hinblättern; Ausweichen auf BASIC spart 200 DM. Unter Hobbyisten, die ihr eigenes Portemonnaie ziehen müssen, ist eine solche Preispolitik nicht sehr angesehen.

Es drängt sich der Eindruck auf, als ob das UCSD-p-System selber an dem Ast sägt, auf dem es sitzt, oder sich in höhere Regionen des Marktes (Minicomputer, Supermikros) flüchtet. Die Beliebtheit des p-Systems läßt jedenfalls bei Homecomputer- und PC-Besitzern deutlich nach. Schade drum.

Die durch Turbo-Pascal und andere 'Billigprodukte' bewiesene Tatsache, daß man Gewinne sogar bei Verkauf von Quelltexten nicht nur über den Preis, sondern auch über Stückzahlen und kluge Verkaufsstrategie erzielen kann, scheint sich immer noch nicht bis in die Chefetagen der meisten Softwarefirmen (zum Beispiel 'Kleinstweich' oder 'Dicketal Risörtsch') herumgesprochen zu haben. Schuld daran sind sicher die Entwicklungskosten für den Kopierschutz des geknackten Kopierschutzes des gekn. . . und all die Käufer, die nach dem Leitspruch 'Es war schon immer etwas teurer, einen besonderen Geschmack zu haben' überhöhte Preise bezahlen.

Aber es scheint auch eine etwas seltsame Auffassung von Geschäftsmoral zu herrschen, denn manche Computerhersteller legen jedem verkauften Computer Programmpakete bei, selbstverständlich lizenziert, die normal gekauft unter Umständen teurer sind als Hard- und Software eines solchen Rechners zusammen. Diese Pakete oder preiswerte rechnergebundene Software-Angebote umfassen mitunter auch das UCSD-p-System, sozusagen für 'nen Appel und 'n Ei.

Soll man sich wirklich eine 'unpassende' Hardware kaufen, nur um billig zu Software-Lizenzen zu kommen? Ach, gäb's doch nur mehr 'Turbos'! c't

Literatur

Karl Achilles, So erstellt man große Programme, Computer Persönlich, Heft 24/85

Kenneth Bowels, Pascal für Mikrocomputer, Springer-Verlag, Berlin/Heidelberg/New York, 1982

Arthur Luehrmann/Herbert Peckham, Apple-II-Pascal, tewi-Verlag, München, 1983

Der Durchbruch von Epromprogrammierern

Gang Programmer Der vorteilhafte Eprom-Kopierer
- Programmiert 2716, 2732, 2732A, 2764, 27128, 27256, EEproms & CMOS.
- Softwarefunktionen einstellbar; keine Extramodule.
- Vorteilhafte Funktionen wie: Fastprogramming, Checksum, Typenanpassung (Master = 2716, Copy = 27256) Überbelastungssicherung usw.
- Kompakte Bauweise.

1195;

Eprom Programmierer/2, universell
- Anschluss an parallel Centronics Port.
- Programmiert 2716, 2732, 2732A, 2764, 27128, 27256, EEproms & CMOS (Option: 27512 oder 2532/2564).
- Anwenderfreundl. Software für: Fastprogramming, Verify, Read, Edit, Dump, Change, Epromtype usw.
- Gleichstartsoftware für: MS-DOS, CP/M, BBC, Apple II, MSX, 68000, CBM64, Z80, u.a.
- Lieferung in solidem Kunststoffgehäuse.

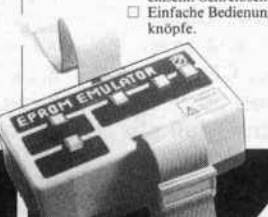
349;

software 75;

Eprom-Löscher
- Für bis zu 4 Eproms.
- In lichtdichtes, kompaktes Gehäuse gebaut.

120;

Eprom-Emulator
- Emuliert 2716, 2732 und 2764, einschl. Schreibschutz.
- Einfache Bedienung durch Druckknöpfe.



495;

ROOS ELEKTRONIK KLEINER MARKT 7 4190 KLEVE TEL: 02821 - 28826

Preise einschl. MwSt.

Name:
Adresse:
Ort + Plz.:
Tel.:

Dokumentation schicken
- Eprom Produkte
- Low Cost Entwickl. Syst.
- Drucker Puffer/Schalter

Deutsche Typen

CPC 464 mit deutschen Zeichen unter CP/M

Andreas Stiller

Umlaute, das 'ß' und vor allem auch das '§' sind die Besonderheiten des deutschen Zeichensatzes, auf die man bei der Textverarbeitung kaum verzichten möchte. Nun sind die CPCs glücklicherweise ausgesprochen flexibel, was die Tastaturbelegung und die Screen-Symbole angeht. Doch neue Zeichen benötigen zusätzlichen Speicherplatz, den beispielsweise WordStar gar nicht mehr erübrigen kann. Dieser Beitrag zeigt, wie man dennoch unter CP/M die neuen Zeichen so einbinden kann, daß sie den Arbeitsspeicher (TPA) unberührt lassen.

Als 'sicheres Örtchen' für die Umbiegeroutine und die Zeichenmatrizen dient, wie auch bei den in c't 1/86 vorgestellten Routinen FAST und ABBLOCK, der unter CP/M wohl kaum genutzte Sound-Puffer. Selbst wenn die beiden genannten Routinen hier bereits untergebracht sind, bleibt noch genügend Platz für die hier vorgestellte Zeichenerweiterung.

Ein kurzes CP/M-Ladeprogramm schiebt das eigentliche Programm, die Zeichenmatrizen und die Tastaturbelegung an die richtigen Stellen und setzt die System-Variablen. Da es durchaus sinnvoll ist, mit dem FAST-Programm die Zeichenausgabe zu beschleunigen, ist diese Routine sogar in einer noch etwas schnelleren Version bereits in die Deutsch-Routine mit eingebaut. Außerdem kann sie nun auch Zeichen invers darstellen (ge'toggle't über das Steuerzeichen &18), was ja beim Turbo-Editor oder bei WordStar die Übersichtlichkeit verbessert. Bei WordStar ist das zugehörige Steuerzeichen in die 'Patch Area' einzutragen (bei Version 3.0 284h:01, 285h: 18h, 28Bh: 01, 28Ch: 18h)

Abgefangen

Genauso wie die FAST-Routine fängt die 'Umbiege'-Routine die Zeichenausgabe an der CPC-Schnittstelle (&BB5A) ab und stellt dann fest, ob es sich um eins der betroffenen Zeichen handelt. Damit hierbei nicht soviel Zeit vertrödelt wird, benutzt sie eine 128 Bytes lange Tabelle, die für jedes ASCII-Zeichen also ein Byte bereithält.

Da unter CP/M normalerweise nur die sieben ASCII-Bits relevant sind, setzt die Routine das achte Bit grundsätzlich auf Null. CPC-Sonderzeichen sind dann allerdings nicht mehr darstellbar. Unter Umständen möchte man aber auf diese Sonderzeichen nicht verzichten, beispielsweise unter Turbo-Pascal. Wenn man die Sonderzeichen partout braucht, kann man vor dem RES 7,A eine kleine Abfrage einschieben:

```
BIT 7,A
JR NZ,ZOUT1
```

Die Tabelle ab &B722 enthält nur für die zu verändernden Zeichen einen Eintrag ungleich Null. Dieser Wert entspricht der Zeichennummer wie bei dem BASIC-Befehl SYMBOL.

Die Zeichenmatrizen – im Handbuch 'Beschreibungswerte' genannt – beginnen ab Label ZMATR. Diese Startadresse wird daher in das zuständige Register der Text-VDU (&B296) eingetragen. Die ersten acht Bitmuster gehören dann zu der Zeichennummer &F0, die nächsten zu &F1 und so weiter. Hier kann sich der Leser auch noch einige eigene Zeichen eintragen, die mit &F9 bis &FF in der Zuordnungstabelle jedes beliebige ASCII-Zeichen 'überschreiben'. Die ersten &20 Bytes der Zuordnungstabelle kann man übrigens bedenkenlos für eigene Zeichenmatrizen nutzen, wenn man auf diejenigen Sonderzeichen verzichtet, die parallel zu den Steuercodes liegen, der Akku also stets > = &20 ist.

Wer sich diese Zuordnungstabelle genau anschaut, wird feststellen, daß auch das 'T' (Code &54) auf eine andere Zeichenmatrix umgebogen ist. Das neue 'T' hat nun unten keinen Fuß mehr. So lassen sich auch auf einem CPC-Farbmonitor 'I' und 'T' mühelos unterscheiden.

Die deutschen Umlaute ä,ö,ü besetzen nach einer internationalen Übereinkunft die Codes für [,] ; \. Diese Codes werden jedoch nicht von der 'CAPS LOCK'-Taste beeinflußt, wie es sonst bei den übrigen Buchstaben der Fall ist. Damit man aber hier nicht sozusagen Buchstaben zweiter Klasse schafft, prüft die 'Verdeutschungs'-Routine zunächst, ob eventuell CAPS LOCK (&B4E8) vorliegt, und setzt gegebenenfalls einen kleingeschriebenen Umlaut in Versalien. Das führt sie aber wirklich nur bei den Umlauten durch (Zeichennummern kleiner &F2), da es zumindest bislang noch kein großgeschriebenes 'ß' oder '§' gibt.

Belegt

Die Tastaturbelegung läßt sich in CP/M mit dem mitgelieferten Install-Programm nach Belieben einstellen. Dennoch ist es zweckmäßig, in das DEUTSCH-Programm eine eigene Tastenbelegung mit aufzunehmen, die die irgendwann installierte überschreibt. Die Bele-

Tastennummer & Taste	Code Normal	Code mit SHIFT	Code mit CTRL
0 Cursor oben	&F0 "p"	&F4 "t"	&F8 :"x"
1 Cursor rechts	&F3 "s"	&F7 "w"	&FB :"{"
2 Cursor unten	&F1 "q"	&F5 "u"	&F9 :"y"
3 9 (Z-Block)	&89 :"9"	&89 :"9"	&89 :"9"
4 6 (Z-Block)	&86 :"6"	&86 :"6"	&86 :"6"
5 3 (Z-Block)	&83 :"3"	&83 :"3"	&83 :"3"
6 ENTER (Z-Bl)	&8B : ^M	&8B : ^M	&8C :run"^M
7 . (Z-Block)	&8A :"."	&8A :"."	&8A :"."
8 Cursor links	&F2 "r"	&F6 "v"	&FA "z"
9 COPY	&E0 "`"	&E0 "`"	&E0 :"`"
10 7 (Z-Block)	&87 :"7"	&87 :"7"	&87 :"7"
11 8 (Z-Block)	&88 :"8"	&88 :"8"	&88 :"8"
12 5 (Z-Block)	&85 :"5"	&85 :"5"	&85 :"5"
13 1 (Z-Block)	&81 :"1"	&81 :"1"	&81 :"1"
14 2 (Z-Block)	&82 :"2"	&82 :"2"	&82 :"2"
15 0 (Z-Block)	&80 :"0"	&80 :"0"	&80 :"0"
16 CLR	&10 ^P	&10 ^P	&10 ^P
17 [	&5B "["	&7B "{"	&1B ^[
18 ENTER	&0D ^M	&0D ^M	&0D ^M
19]	&5D "]"	&7D "}"	&1D ^]
20 4 (Z-Block)	&84 :"4"	&84 :"4"	&84 :"4"
21 SHIFT	&FF	&FF	&FF
22 \	&5C "\"	&60 "`"	&1C ^\
23 CTRL	&FF	&FF	&FF
24 ^	&5E "^"	&A3 "#"	&1E ^^
25 -	&2D "-"	&3D "="	&FF
26 @	&40 "@"	&7C "|"	&00 ^@
27 P	&70 "p"	&50 "P"	&10 ^P
28 ;	&3B ";"	&2B "+"	&FF
29 :	&3A ":"	&2A "*"	&FF
30 /	&2F "/"	&3F "?"	&FF
31 .	&2E "."	&3E ">"	&FF
32 0	&30 "0"	&5F "_"	&1F ^_
33 9	&39 "9"	&29 ")"	&FF
34 O	&6F "o"	&4F "O"	&0F ^O
35 I	&69 "i"	&49 "I"	&09 ^I
36 L	&6C "l"	&4C "L"	&0C ^C
37 K	&6B "k"	&4B "K"	&0B ^B
38 M	&6D "m"	&4D "M"	&0D ^D
39 ,	&2C ","	&3C "<"	&FF
40 8	&38 "8"	&28 "("	&FF
41 7	&37 "7"	&27 "'"	&FF
42 U	&75 "u"	&55 "U"	&15 ^U
43 Y	&79 "y"	&59 "Y"	&19 ^Y
44 H	&68 "h"	&48 "H"	&08 ^H
45 J	&6A "j"	&4A "J"	&0A ^J
46 N	&6E "n"	&4E "N"	&0E ^E
47 Leertaste	&20 " "	&20 " "	&FF
48 6 /J1 oben	&36 "6"	&26 "&"	&FF
49 5 /J1 unten	&35 "5"	&25 "%"	&FF
50 R /J1 links	&72 "r"	&52 "R"	&12 ^R
51 T /J1 rechts	&74 "t"	&54 "T"	&14 ^T
52 G /J1 Feuer2	&67 "g"	&47 "G"	&07 ^G
53 F /J1 Feuer1	&66 "f"	&46 "F"	&06 ^F
54 B	&62 "b"	&42 "B"	&02 ^B
55 V	&76 "v"	&56 "V"	&16 ^V
56 4	&34 "4"	&24 "$"	&FF
57 3	&33 "3"	&23 "#"	&FF
58 E	&65 "e"	&45 "E"	&05 ^E
59 W	&77 "w"	&57 "W"	&17 ^W
60 S	&73 "s"	&53 "S"	&13 ^S
61 D	&64 "d"	&44 "D"	&04 ^D
62 C	&63 "c"	&43 "C"	&03 ^C
63 X	&78 "x"	&58 "X"	&18 ^X
64 1	&31 "1"	&21 "!"	&FF
65 2	&32 "2"	&22 """	&7E "~"
66 ESC	&FC "|"	&FC "|"	&FC "|"
67 Q	&71 "q"	&51 "Q"	&11 ^Q
68 TAB	&09 ^I	&09 ^I	&E1 "a"
69 A	&61 "a"	&41 "A"	&01 ^A
70 CAPS LOCK	&FD	&FD	&FE
71 Z	&7A "z"	&5A "Z"	&1A ^Z
72 J0 oben	&0B ^K	&0B ^K	&FF
73 J0 unten	&0A ^G	&0A ^G	&FF
74 J0 links	&08 ^E	&08 ^E	&FF
75 J0 rechts	&09 ^F	&09 ^F	&FF
76 J0 Feuer2	&58 "X"	&58 "X"	&FF
77 J0 Feuer1	&5A "Z"	&5A "Z"	&FF
78 frei	&FF	&FF	&FF
79 DEL	&7F DEL	&7F DEL	&7F DEL

So belegt das Betriebssystem nach einem Reset die Tastatur auf allen drei Ebenen.

gung stimmt dann immer! Zudem ist sie in dem hier angeführten Beispiel besonders auf das Arbeiten mit WordStar abgestimmt. Damit keine Totalkonfusion auftritt, verbleiben aber 'Y' und 'Z' auf ihren DIN-feindlichen ASCII-Plätzen (QWERTY). Auch die nach DIN genormten Positionen von '<','>' und so weiter bleiben beharrlich dort, wo sie beim CPC aufgedruckt sind.

Die Belegungstabellen beginnen beim CPC 464 ab &B34C. Sie ordnen jeder Taste drei Codes zu, und zwar jeweils einen für normal (ab &B34C), mit SHIFT (ab &B39C) und mit CTRL (ab &B3EC). Die Reihenfolge der Tasten ist im Anhang III des Handbuchs aufgeführt, beginnend mit Taste 0 (aufgedruckt: '↑') bis Taste 79 ('DEL'), wobei zwölf Tastennummern den Joysticks zugeordnet sind.

Der Belegungs-Code &FF hat die Bedeutung: keine Wirkung. Ansonsten entsprechen die Codes &00 bis &1F den Steuerzeichen, &20 bis &7F den ASCII-Zeichen. Die Bedeutung der Codes &A0 bis &FE richtet sich nach der 'Umgebung': in BASIC sind dies CPC-Sonderzeichen, Cursor-Befehle oder Escape (&E0,&E1,&EF bis &FE).

Unter CP/M wird bei diesen Codes das achte Bit gelöscht, so daß beispielsweise der Taste mit '↑' mit der Normalbelegung &F0 das ASCII-Zeichen 'p' (&70) entspricht.

Erweitert

Die Codes zwischen &80 und &9F haben, wie man es von der BASIC-Funktion KEY her kennt, eine Sonderfunktion. Hiermit lassen sich Ausgabe-Strings kennzeichnen, die in den KEY-Erweiterungspuffer (ab &B446) abzulegen sind. Der Code &80 gehört zu dem ersten String, &81 zum zweiten und so weiter. Jedem String muß dabei ein Längenbyte vorangehen. Insgesamt stehen für alle Strings

Tastennummer & Taste	Code Normal	Code mit SHIFT	Code mit CTRL
0 Cursor oben	&05 ^E	&05 ^E	&8C :^QE
1 Cursor rechts	&04 ^D	&06 ^F	&8D :^QD
2 Cursor unten	&18 ^X	&18 ^X	&8E :^QX
6 ENTER (Z-Bl)	&8B :^M	&8B :^M	&8B :^M
8 Cursor links	&13 ^S	&01 ^A	&8F :^QS
9 COPY	&90 :^KS	&90 :^KS	&91 :^KS^QP
13 1 (Z-Block)	&81 :"1"	&81 :"1"	&93 :Mit freundlichen Grüßen^M^M
15 0 (Z-Block)	&80 :"0"	&80 :"0"	&92 :Sehr geehrte Damen und Herren,^M^M
16 CLR	&13 ^S	&13 ^S	&13 ^S
17 [	&7B "ä"	&5B "Ä"	&1B ^[
19]	&7D "ü"	&5D "Ü"	&1D ^]
22 \	&7C "ö"	&5C "Ö"	&1C ^\
24 ^	&7E "ß"	&5E "^"	&1E ^^
26 @	&40 "§"	&60 "`"	&00 ^@
65 2	&32 "2"	&22 """	&FF
66 ESC	&1B ^[	&1B ^[	&FC "|"
68 TAB	&09 ^I	&09 ^I	&09 ^I

Für die deutschen Zeichen ist unter CP/M eine Umbelegung sinnvoll. Gleiches gilt für die Cursor-Tasten und eventuelle Erweiterungsstrings für oft benötigte Befehle oder Floskeln.

aber nur 100 Bytes in dem Puffer zur Verfügung.

Wer unbedingt mehr braucht, kann den Puffer irgendwohin in den freien Speicher verlegen und die neue Pufferadresse dem Keyboard Manager mitteilen (nach &B4E1). Ein Tip: wenn man nicht mit Kassetten arbeitet, kann man dafür den Kassettenpuffer (&B800 bis &B8FF) nutzen.

Die Codes von &80 bis &89 bleiben auch unter CP/M den Ziffern 0 bis 9 zugeordnet (für den Ziffernblock). Die kleine ENTER-Taste erhält dieselbe Bedeutung wie ENTER, so daß die Codes von &8A aufwärts für sinnvolle Textverarbeitungs-Strings zur Verfügung stehen. Vor allen Dingen für die Pfeiltasten, Delete und Copy bietet es sich an, zweckmäßige Strings zu definieren. Das hier vorgestellte Programm belegt diese Tasten WordStar-gerecht, um den Cursor zu positionieren, Zeichen zu löschen oder eine Datei abzuspeichern.

Die Pfeiltasten positionieren den Cursor dann normalerweise buchstabenweise, mit SHIFT wortweise und mit CTRL immer an den Rand des Bildes. Copy zusammen mit CTRL gibt die Sequenz ↑KS↑QP aus, so daß der Cursor nach dem Abspeichern wieder seine alte Position einnimmt. Bei der Programmierung dieses Strings zeigte sich allerdings ein 'Bug' im Betriebssystem: Beginnt der String mit einem Steuerzeichen und ist er länger als zwei, so 'verschluckt' der Keyboard Manager das zweite Zeichen. Daher wurde der String um ein Dummy-Zeichen (hier &00) verlängert.

Als Beispiel, wie man auch dauernd benötigte Floskeln zuordnen kann, sind die Strings 'Sehr geehrte Damen und Herren, <CR><CR>' unter &92 und 'Mit freundlichen Grüßen' <CR><CR> unter &93 im Erweiterungspuffer untergebracht. In der Belegungstabelle erhalten CTRL 0 und CTRL 1 (im Ziffernblock) diese Belegung.

Eine wichtige Änderung gegenüber der Normalbelegung ist der Code der ESC-Taste (das fallende Bömbchen).

Sie bekommt den Standardwert &1B zugewiesen. Manche speziell für den Schneider angepaßten Programme, insbesondere Turbo-Pascal, benötigen aber den 'Bömbchen-Code' &FC für 'Escape'. Hier sollte man aber lieber mit einem gezielten Patch die Programme wieder auf den Standardwert zurückbiegen; bei Turbo 3.0 ist dazu die Adresse &2E8B auf &1B zu setzen.

Als 'Notausstieg' verbleibt aber der alte Code auf der ESC-Taste, wenn man sie zusammen mit CTRL drückt (Achtung! Zusammen mit SHIFT ergibt sich der totale Notausstieg).

Als letzte Tabelle sind noch die zehn Bytes der 'Repeat Map' interessant. Hier ist für alle 80 Tasten vermerkt, ob nach einer kurzen Wartezeit eine weiterhin gedrückte Taste zu einer Zeichenwiederholung führen soll (Bit =1) oder nicht (Bit =0). Das erste Bit (von rechts!) ist Taste 0 zugeordnet, das zweite der Taste 1 und so weiter.

Die &07 im ersten Byte bedeutet also, daß die Pfeile (0,1,2 alle drei Bits auf 1) zu wiederholen sind, die Tasten '9','6','3','Enter' und '.' auf dem Zifferblock mit den Tastennummern drei bis sieben jedoch nicht.

Gegenüber der Normalbelegung wird lediglich das zur Tastennummer neun gehörige Bit geändert, da es nicht gerade sinnvoll ist, ein 'Dauer-Copy' auszugeben.

Übrigens sind beim Joystick 0 die beiden Feuerknöpfe (Tastennummern 76 und 77) auf 'Einzelfeuer' gestellt. Wer aus welchen Gründen auch immer 'Dauerfeuer' benötigt, braucht sich kein Spezial-Interface dafür zu kaufen (gibt's wirklich). Es reicht, wenn man das Byte &B445 auf &FF setzt.

Zurückschalten

Ein Mini-Programm reicht aus, um wieder die ASCII-Zeichen zu installieren:

```
100 LD HL,134Ah
103 LD (BDD4h),HL
106 JP 0000
```

Die gleiche Wirkung hat auch der Aufruf der FAST-Routine aus c't 1/86, die aber zusätzlich die Zeichenausgabe beschleunigt.

Nur hat man sich jetzt an die leicht geänderte Tastaturbelegung zu gewöhnen, das Pfund-Symbol fehlt und hat statt dessen '↑', auf '↑' liegt die Tilde, über '\' liegt '|' und über '@' liegt '`'. Wer jedoch die alte Belegung lieber hat – ein Kaltstart macht's möglich (CTRL/SHIFT/ESC und ICPM).

```
 .z80
CSEG

;*****************************************************
;CPC 464 mit deutschen Zeichen:(äöüÄÖÜß@)
;und 'T' ohne Fuß, um 'T' und 'I' besser
;unterscheiden zu können, mit integrierter FAST-
;Routine
;*****************************************************
        ORG     100H

WBOOT   EQU     0       ;Warmboot

;       Systemadressen des CPC 464

KNORM   EQU     0B34CH  ;Beginn Tastaturbelegung
CAPSL   EQU     0B4E8H  ;Flag für CAPS LOCK
WRCHAR  EQU     0BDD4H  ;Vektor für Zeichenausg.
STMATR  EQU     3800H   ;Standard Zeichenmatrix
CHPOS   EQU     0B64H   ;Bestimmt Char-Position
PZIEL   EQU     0B680H  ;Adresse der Umbiegerout.
INVERS  EQU     0B290H  ;Flag für PEN/PAPER
;*****************************************************
;Lader
;*****************************************************

START:  LD      DE,PZIEL        ;Verschiebt Umbiege-
        LD      HL,ZUMBG        ;programm
        LD      BC,ZUMEND-ZUMBG
        LDIR
VER     EQU     ZUMBG-PZIEL     ;Verschiebung
        LD      HL,PZIEL        ;verbiegt Ausgabe-
        LD      (WRCHAR),HL     ;Vektor
        LD      HL,TNORM        ;lädt neue
        LD      DE,KNORM        ;Tastaturbelegung
        LD      BC,TEND-TNORM
        LDIR
        JP      WBOOT           ;und Abgang

;*****************************************************
;Umbiege-Programm
;*****************************************************

ZUMBG:  PUSH    HL              ;rettet HL
        LD      HL,ZTAB-VER     ;HL weist auf Tabelle
        LD      D,0
;****    BIT     7,A     ***     ;Nur falls
;****    JR      NZ,ZOUT1***     ;Sonderzeichen erwünscht
        RES     7,A             ;Löscht immer 8. Bit
        LD      E,A             ;Berechnet Zeichenadresse
        ADD     HL,DE           ;in Tabelle
        BIT     7,(HL)          ;gibt es einen Eintrag
        JR      Z,ZOUT1         ;nein, dann Ausgabe
        LD      A,(HL)          ;hole "SYMBOL"
        SUB     0F0H            ;relativ zu 0F0H
        LD      HL,CAPSL        ;CAPS LOCK?
        BIT     7,(HL)
        LD      HL,ZMATR-VER    ;Neue Matrix
        JR      Z,ZOUT2         ;nein, dann Ausgabe
        CP      03H             ;kleiner Umlaut?
        JR      NC,ZOUT2        ;nein, dann Ausgabe
        ADD     A,3             ;sonst verwandeln
        JR      ZOUT2           ;und ausgeben
ZOUT1:  LD      HL,STMATR       ;Standardmatrix
ZOUT2:  LD      E,A             ;DE: Zeichencode
        EX      DE,HL           ;und nach HL
        ADD     HL,HL           ;Bestimmt Matrix
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,DE           ;Matrix-Start nach
        EX      DE,HL           ;DE
        POP     HL              ;H=Spalte, L=Zeile
        CALL    CHPOS           ;Bestimmt  Adresse
        LD      A,(INVERS)      ;C:=Inversflag
        LD      C,A
        LD      B,8             ;B:=8 Zeilen
NXTZL:  LD      A,(DE)          ;Matrix-Zeile nach A
        XOR     C               ;eventuell invers
        LD      (HL),A          ;und ins Bild
        LD      A,H             ;Bildadresse um
        ADD     A,8             ;8*256 Erhöhen
        LD      H,A
        INC     DE              ;nächste Matrixzeile
        DJNZ    NXTZL           ;bis 8
        RET                     ;fertig


;*****************************************************
;Zeichenmatrizen je acht Bytes pro Zeichen
;*****************************************************

ZMATR:  DB      66h,00h,78h,0Ch,7Ch,0CCh,76h,00h    ;"a"
        DB      66h,00h,3Ch,66h,66h,66h,3Ch,00h     ;"ö"
        DB      66h,00h,00h,66h,66h,66h,3Eh,00h     ;"ü"
        DB      0DBh,3Ch,66h,66h,7Eh,66h,66h,00h    ;"A"
        DB      0D6h,6Ch,0C6h,0C6h,0C6h,6Ch,38h,00h;"Ö"
        DB      66h,00h,66h,66h,66h,66h,3Ch,00h     ;"Ü"
```

```
        DB      7Eh,5Ah,18h,18h,18h,18h,18h,00h     ;"T"
        DB      38h,6Ch,6Ch,6Ch,66h,76h,6Ch,60h     ;"ß"
        DB      1Ch,32h,38h,6Ch,38h,98h,70h,00h     ;"§"
        DEFS    18H
        ;Platz für weitere Zeichenmatrizen

;*****************************************************
;Tabelle enthält für jedes ASCII-Zeichen einen Eintrag
;Das erspart reichlich Rechen- und Suchzeit
;*****************************************************

ZTAB:   DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      0F8h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,0F6h,00h,00h,00h
        DB      00h,00h,00h,0F3h,0F4h,0F5h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,00h,00h,00h,00h,00h
        DB      00h,00h,00h,0F0h,0F1h,0F2h,0F7h,00h
ZUMEND:

;*****************************************************
;Tastaturbelegung
;*****************************************************

TNORM:  DB      05h,04h,18h,89h,86h,83h,8Bh,8Ah
        DB      13h,90h,87h,88h,85h,81h,82h,80h
        DB      13h,7Bh,0Dh,7Dh,84h,0FFh,7Ch,0FFh
        DB      7Eh,2Dh,40h,70h,3Bh,3Ah,2Fh,2Eh
        DB      30h,39h,6Fh,69h,6Ch,6Bh,6Dh,2Ch
        DB      38h,37h,75h,79h,68h,6Ah,6Eh,20h
        DB      36h,35h,72h,74h,67h,66h,62h,76h
        DB      34h,33h,65h,77h,73h,64h,63h,78h
        DB      31h,32h,1Bh,71h,09h,61h,0FDh,7Ah
        DB      08h,0Ah,08h,09h,58h,5Ah,0FFh,7Fh

TSHIFT: DB      05h,06h,18h,89h,86h,83h,8Bh,8Ah
        DB      01h,90h,87h,88h,85h,81h,82h,80h
        DB      13h,5Bh,0Dh,5Dh,84h,0FFh,5Ch,0FFh
        DB      5Eh,3Dh,60h,50h,2Bh,2Ah,3Fh,3Eh
        DB      5Fh,29h,4Fh,49h,4Ch,4Bh,4Dh,3Ch
        DB      28h,27h,55h,59h,48h,4Ah,4Eh,20h
        DB      26h,25h,52h,54h,47h,46h,42h,56h
        DB      24h,23h,45h,57h,53h,44h,43h,58h
        DB      21h,022h,1Bh,51h,09h,41h,0FDh,5Ah
        DB      08h,0Ah,08h,09h,58h,5Ah,0FFh,7Fh

TCTRL:  DB      8Ch,8Dh,8Eh,89h,86h,83h,8Bh,8Ah
        DB      8Fh,91h,87h,88h,85h,93h,82h,92h
        DB      13h,1Bh,0Dh,1Dh,84h,0FFh,1Ch,0FFh
        DB      1Eh,0FFh,00h,10h,0FFh,0FFh,0FFh,0FFh
        DB      1Fh,0FFh,0Fh,09h,0Ch,08h,0Dh,0FFh
        DB      0FFh,0FFh,15h,19h,08h,0Ah,0Eh,0FFh
        DB      0FFh,0FFh,12h,14h,07h,06h,02h,16h
        DB      0FFh,0FFh,05h,17h,13h,04h,03h,18h
        DB      0FFh,0FFh,0FCh,11h,09h,01h,0FEh,1Ah
        DB      0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,7Fh

;Repeat Map

RPMAP:  DB      07h,01h,4Bh,0FFh,0FFh,0FFh,0FFh,0FFh
        DB      0ABh,8Fh

;Stringerweiterung

STERW:  DB      1,'0',  1,'1',  1,'2',  1,'3'
        DB      1,'4',  1,'5',  1,'6',  1,'7'
        DB      1,'8',  1,'9',  1,'.',  1,0DH

        DB      2,11h,'E',  2,11h,'D',  2,11h,'X'
        DB      2,11h,'S',  2,0Bh,'S'

        DB      5,0Bh,00,'S',11h,'P'

;zusätzliche Beispielstrings (ab Nummer &92)

        DB      32,'Sehr geehrte Damen und Herren,'

        DB      0Dh,0Dh
        DB      25,'Mit freundlichen Grüßen',0Dh,0Dh
TEND:
        END
```

Ein Lader schiebt das Deutsch-Programm an die richtige Stelle. Der Verschiebewert "VER" muß immer als Differenz in den Berechnungen auftauchen, falls der M80-Assembler mit CSEG arbeitet. c't

MicroProcessing

Programmer der Reihe DIAGNOSTIC diagnostizieren Ihnen bereits heute Fehler, die sonst erst morgen auftreten würden.

EPROM PROGRAMMER

Der meistverkaufte Programmer des Jahres 1986

★ DIAGNOSTIC 512 ★

Für Labor und Service: 1 MASTER + 4 Kopien

neu sensationell: ab sofort sind EPROM-Typen bis '512 und RAM-Speicher von 512 Kbit im Preis von 3985 DM + MwST. inbegriffen.

neu schnellste Programmier-Algorithmen, z. B. Quick Puls Programming von INTEL in 4 Sekunden P27256!

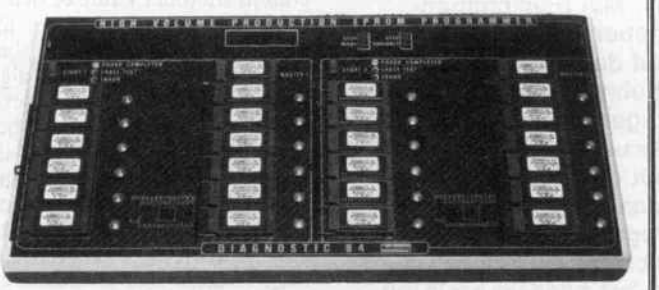
neu APP (advanced programming package) für höchste Ansprüche, unterstützt z. B. anwenderprogrammierbare verschärfte Vcc margins usw.

neu CPP (communicator programm package) unterstützt menügesteuertes Dialog mit IBM-PC/XT

neu Adapter für löschbare PALs (LATTICE) und 1 Mbit EPROMs

Die weltkostengünstigste Art, EPROMs zu programmieren und dynamisch (Zugriffszeit!) zu testen

★ DIAGNOSTIC 84 ★



Für großvolumige Massenproduktion: 2 MASTER + 24 Kopien
Für mittlere Produktion : 1 MASTER + 1 Kopie

UV-Schnell-Löschgerät
für 30 EPROMS IN 5—10 MIN

Hannover Messe '86
9.—16. 4. 1986
Halle 12, 2. OG., St. 2013

MICROPROCESSING GMBH,
AM VOGELHERD 13, 8000 MÜNCHEN 71,
TEL. 0 89/7 59 19 36

Deutsche Steuern

CPC berechnet menügesteuert die Lohn-/Einkommensteuer 1985

Michael Hanke

Alle Jahre wieder, immer wenn es auf den 31. Mai (Einkommensteuererklärung) oder auf den 30. September (Lohnsteuerausgleich) zugeht, müssen sich die Steuerpflichtigen mit dem Fiskus auseinandersetzen. Der begnügt sich aber beileibe nicht mit dem klassischen Zehnt, sondern greift mit Hilfe komplizierter Steuerformeln bei steigendem Einkommen immer tiefer in die Taschen der arg gebeutelten Bürger. Meist haben nur Fachleute die Chance, da durchzublicken – doch mit Hilfe eines menügesteuerten Programms kann sich auch jeder (arbeitnehmende) Laie ausrechnen, was 1986 auf ihn zukommt.

Man nehme sich nur ein Steuerformular, tippe das Programm ein (oder kaufe es sich – steuerlich absetzbar – im c't-Software-Service) und beantworte die in der Reihenfolge der Formular-Randnummern gestellten Fragen. Dabei beschränkt sich das Programm allerdings auf die Einkünfte aus nichtselbständiger Arbeit (Anlage N) sowie auf die Einkünfte aus Vermietung und Verpachtung (Anlage V). Selbständige müssen sich also nach wie vor einen Steuerberater leisten.

Auch wer Tips und Schlichen zur Steuerersparnis erfahren möchte, sollte einen Steuerberater konsultieren, da das hierfür nötige Expertensystem leider noch nicht entwickelt ist (ein gutes Buch tut's auch, beispielsweise der 'Renner' von Konz: '1000 ganz legale Steuertricks').

Kontrolle

Im Unterschied zu einem Finanzbeamten erkennt das Programm alle Angaben, wie Werbungskosten, Fahrtkosten und so weiter, als gegeben an. Wer will, kann zusätzlich bei einigen Posten mit einem Zufallsgenerator (in etwa 50% zu 50%) mehr 'Wirklichkeitsnähe' erreichen.

Als Ergebnis erhält man – wahlweise auf Drucker oder Bildschirm – alle wichtigen Informationen wie Werbungskosten, Vorsorgebeträge und vor allem die Höhe der zu erwartenden Steuererstattung (hoffentlich) beziehungsweise der Nachzahlung (oh Graus).

Sollte die Finanzamtsberechnung von der eigenen abweichen, wobei dafür nur die paar Beträge anzugeben sind, die behördlich anerkannt wurden, so ist ein gesundes Mißtrauen angebracht. Fällt die offizielle Berechnung gar ungünstiger aus (auch Ämter können mal irren), sollte man die ganze Abrechnung noch einmal sorgfältig kontrollieren und gegebenenfalls flugs Einspruch (bei Kirchensteuer heißt das Widerspruch) einlegen. Man hat für den Rechtsbehelf nur einen Monat Frist.

Aber auch mit der Ablehnung von einigen Werbungskosten oder Sonderausgaben braucht man sich nicht zufriedenzugeben. Ein formloser Einspruch hat hier schon so manche Mark vor dem Fiskus gerettet (wer sich nicht wehrt. . .).

Die Spirale

Wie nun die Steuern tatsächlich berechnet werden, bleibt oft auch Fachleuten verborgen (von wegen Transparenz im Steuerwesen). Hierfür gibt es Tabellen, die aber auch nicht gottgegeben sind, sondern die man aus merkwürdigen Formeln zusammengebraut hat.

Allein schon die Rundungsmethode der fiskalischen Mathematiker mutet etwas seltsam an. Gerundet wird nämlich auf Vielfaches der geheimnisvollen Zahl 54:

E-rund = 54 * INT (E/54)

Da also grundsätzlich abgerundet wird, sollte man mit seinem Chef immer ein Jahres-Bruttoeinkommen aushandeln (über das Weihnachtsgeld), so daß das zu versteuernde Einkommen 54*n+53 beträgt. Und dieses Programm dient schließlich dazu, das zu versteuernde Einkommen zu berechnen. Unter Umständen führt eine Mark Brutto weniger dazu, rund 25 DM an Steuern zu sparen – immerhin!

Der abgerundete Wert des zu versteuernden Einkommens wird in eine von fünf Steuerzonen eingestuft (Grundtabelle): Unter 4266 DM entfällt jegliche Steuer, bis 18000 DM kommt die untere Proportionalzone, bis 60000 die untere Progressionszone, bis 130000 DM die obere Progressionszone und darüber die obere Proportionalzone mit dem Höchststeuersatz von 56%. In den Proportionalzonen steigt der Steuerbetrag linear mit dem Einkommen an, in den Progressionszonen sorgt jeweils ein Polynom vierter Ordnung dafür, daß es mit dem Steuersatz schnell bergauf geht.

Die Zonen sind geschickterweise so angelegt, daß sie recht nahtlos ineinander übergehen. Beim Steuersatz (Steuerbetrag zu Einkommen) muß man die Freibeträge berücksichtigen, die auch in der Proportionalzone zu einem zusätzlichen Anstieg (Hyperbelfunktion) führen.

Da wohl kaum jemand den Verlauf der Progressionszonen anhand der nebenstehenden Formeln überblicken kann, ist ein 'Plot' der streng monoton steigenden Steuersatzfunktion aufschlußreich.

Die Steuerzonen

0 - 4165 steuerfrei

4166 - 17999 untere Proportionalzone
Steuer = 0.22*Ein-926

18000 - 59999 untere Progressionszone
Steuer = $3.05*a^4-73.76*a^3+695*a^2+2200*a+3034$
mit a = (Ein-18000)/1000

60000 - 129999 obere Progressionszone
Steuer = $0.09*a^4-5.45*a^3+88.13*a^2+5040*a+20018$
mit a = (Ein-60000)/10000

ab 130000 obere Proportionalzone
Steuer = 0.56*Ein-14837

pc-plus GmbH
Effnerstr. 38 · D-8000 München 81
Telefon 089 / 98 54 66 · Telex 5216948 pcpl d

BASIC pc + dACCESS III Datenbank

Mit pc + dACCESS III wird jetzt der direkte Zugriff über Schlüsselfelder (Index) aus **BASIC** (Interpreter, Compiler) möglich. Eine einfache CALL-Schnittstelle verbindet die Programmiersprache mit **pc + dACCESS III.** Standardfunktionen erlauben vorwärts und rückwärts blättern mit oder ohne Index, ändern und löschen der Datenbanksätze. Weitere interaktive Funktionen unterstützen das Listen und Ändern von Sätzen ohne zusätzlichen Programmieraufwand. Die Datenbank kann auch unverändert von **dBASE III** weiter verarbeitet werden. (100% kompatibel). **pc + dACCESS III** ist für alle gängigen Programmiersprachen verfügbar. (deutsche Dokumentation)

COUPON

Wir bestellen hiermit:

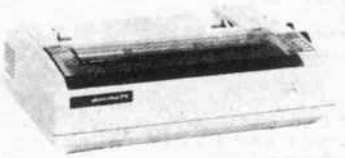
☐ Eine kostenlose Information über pc + dACCESS III
☐ pc + dACCESS III — Demoversion (57,— DM, incl. MwSt.)
☐ pc + dACCESS III — Produkt (1.014,60 DM, incl. MwSt.)
(Nur für IBM und 100% kompatible unter PC-DOS)
Bitte Programmiersprache und Version unbedingt angeben:

Programmiersprache _____ Version _____

Firma _____ Name _____

☐ Scheck liegt bei Straße _____ PLZ/Ort _____

☐ Nachnahme Unterschrift _____ Telefon _____



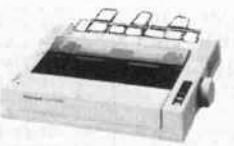
NEC-Pinwriter P2/P3

Der ideale Drucker für hervorragende Schriftqualität mit 21 x 24 Matrix. Schnelldruck mit 180 Z/Sek. Ladbarer Zeichensatz mit 16 x 12 Matrix. Grafikausdruck in bester Auflösung. Viele Schriften zur Wahl, voll IBM/Centronics kompatibel. An fast jeden Computer anschließbar.

Wir liefern Ihnen den NEC Pinwriter P2/P3 **komplett** mit **Centronics/IBM komp. Schnittstelle, Papiertraktor** und **eingebautem Einzelblatteinzug** zum **SONDERPREIS** einschl. MwSt.:

PINWRITER P2 — 80 Zeichen/Zeile nur 1690,— DM
PINWRITER P2 — Color nur 2480,— DM
PINWRITER P3 — 136 Zeichen/Zeile nur 2490,— DM
NEC-Pinwriter P5 (Datenblatt anford.) nur 3950,— DM
Interface für Apple II, C64, CBM, Aufpreis 120,— DM

Wir liefern alle NEC-Matrix- und Typendrucker/auch Farbe, zu günstigen Preisen. Gerne senden wir Ihnen unser Angebot mit informationsmaterial und Probeausdruck. — Händleranfragen willkommen.



Panasonic KX-P1080

Der neue Panasonic Drucker mit Leistungsdaten, die Sie bei anderen Druckern kaum finden werden:

Sechzehn verschiedene NLQ-Schriften, auch in Kursivschrift (Italic). **Matrix (NLQ) 18 x 18 Punkte!** Zeichen je Zeile: 40, 48, 66, 80, 96, 132. Max. 100 Z/Sek. Natürlich auch Proportionalschrift und Sub-Superscript. Grafikpunkte je Zeile: 480, 576, 640, 720, 960, 1920. Ladbarer Zeichensatz! Formatbefehle für Blocksatz u. zentrieren. Verstellbarer Traktor und Walze für Einzelblatt. Druckkopf Lebensdauer 100 Mill. Zeichen. Farbband autom. Nachtränkung für 3 Mill. Zeichen. Leiser Betrieb! Eingebaute Centronics-Schnittstelle.

Panasonic KX-P 1080 100 Z./s ..... nur 875.– DM
Panasonic KX-P 1091 120 Z./s ..... nur 945.– DM
Panasonic KX-P 1092 180 Z./s ..... nur 1295.– DM
Panasonic KX-P 1592 – Breitformat – . nur 1895.– DM
Aufpreis für Grafikinterf. m. Kabel .... nur 120.– DM
(für C64, C128, Apple II)

DIE NEUEN COMMODORE- U. SCHNEIDER-PCs:

CPC 464 mit Grünmonitor: **698.–** mit Farbmonitor: **1050.–**
CPC 664 mit Grünmonitor: **975.–** mit Farbmonitor: **1695.–**
CPC 6128 m. Grünmonit.: **1435.–** mit Farbmonitor: **1895.–**
Commodore PC 128 – jetzt sofort lieferbar! . . **nur 835.–**
Commodore Floppy 1571, 360 KByte für PC 128 . **835.–**
Commodore PC 128-D, mit eingeb. Floppy 360 K **1745.–**
Commodore C64 – noch immer interessant – **465.–**
Commodore Floppy 1541 – neue Ausführung **545.–**
Einzelfloppy SFD 1001 1 Megabyte (IEC-Bus) **nur 795.–**
Doppelfloppy CBM 8250 LP **3290.–**
A3 Flachplotter KPL-710 6farb., HP-GL kompat. . . **2690.–**

Commodore PC 10, kpl. mit 2× 360 KByte Floppy, 12"-Monitor, 256 KByte Speicher, DIN-Tastatur . . **4190.–**
Speicheraufrüstung PC 10/20 auf 512 KByte **275.–**
Speicheraufrüstung PC 10/20 auf 640 KByte **395.–**
Color-Grafikkarte RGB und Videoausgang **345.–**
Grafikkarte 720 × 384 Punkte/Herkules komp. **695.–**
Uhrenkarte mit Kalender **198.–**

Wir liefern auch NEC und TANTON Festplattenlaufwerke sowie alle Computer von COMMODORE, TANTON und VICTOR!
Fragen Sie nach unseren günstigen Angeboten, wenn Sie sich einen PC oder Homecomputer kaufen wollen.

Gerne senden wir Ihnen auch ausführliche Einzelinformationen. Auch Händleranfragen sind uns willkommen.

T. WEBER ELEKTRONIK, 8700 WÜRZBURG, Eisenbahnstraße 22, Tel. 09 31/70 14 41

Wir haben nicht nur den Computer,
wir haben auch die entsprechende Software:

– Crossreferenzprogramm f. BASIC	65,– DM
– Diskmonitor	65,– DM
– Dateiverwaltung	490,– DM
– Adressverwaltung	490,– DM
– Textverarbeitung	120,– DM
– Finanzbuchhaltung	700,– DM
– Lohnprogramm	640,– DM
– Auftragsbearbeitung	655,– DM
– Bestellwesen	600,– DM
– Branchenpakete je nach Umfang	

Wir beraten Sie gern.
Telefon 0 80 31-6 80 21

SZEREDY COMPUTING

SZEREDY COMPUTING GmbH · Theodor-Gietl-Str. 3 · 8200 Rosenheim

MEGA-ATARI

PERSONALCOMPUTER DES JAHRES
1985

Im Programm-Listing sind die Berechnungsformeln in den Zeilen 2260 bis 2370 zu finden. Hier muß man – neben Änderungen bei den Kinder-Freibeträgen – die Formeln in den nächsten Jahren der großen Steuerreform anpassen.

Spitzensteuer

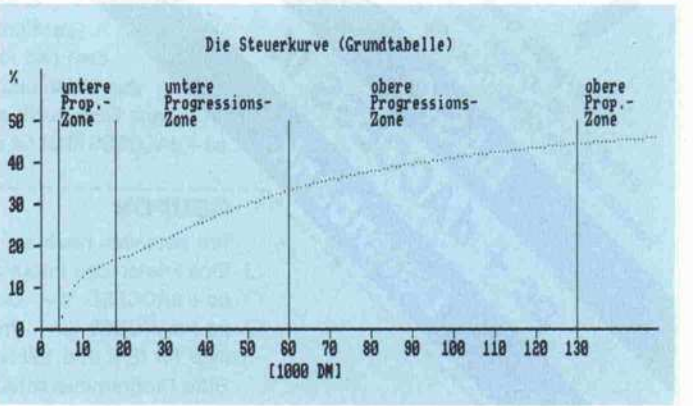
Wichtig ist oftmals aber gar nicht der absolute Steuersatz, sondern der differentielle, im Steuerjargon auch Spitzensteuersatz genannt. Damit kann man überblicken, wieviel Steuern man spart, wenn man 54 DM weniger zu versteuern hat, oder auch was man an Steuern draufzahlt, bei einer Erhöhung um 54 DM.

Diff-E =
(Steuer (E) - Steuer (E-54))/54

Die Spitzensteuersätze liegen bereits bei einem zu versteuernden Einkommen von 34000 DM bei 40% (absolut 23,6%) und bei 52000 DM mit 50% (absolut 30,8%) schon nahe an dem Höchststeuersatz von 56%.

Wer mit 52000 DM gesegnet ist und 100 DM zusätzlich absetzen kann, erhält demnach vom Finanzamt 50 DM 'bar cash' zurück.

Die hohen differentiellen Steuersätze zeigen auch, wie gut der Staat an einer allgemeinen Einkommensverbesserung verdient. Steigt der Lohn beispielsweise um 3,5% von 34000 DM auf 35190 DM, so gehen rund 476 DM zusätzlich an den Fiskus, was dem Staat eine 'Rendite' von 5,9% einbringt. Außerdem bekommt er bei einer Gehaltserhöhung im öffentlichen Dienst gleich rund 40% an Steuern retour, so daß die Mehrbelastung nicht 3,5%, sondern nur 2% ausmacht.



Bei steigendem Jahreseinkommen wächst der Steueranteil und wächst und wächst . . . und der Fiskus 'sahnt ab'.

```
Lohnsteuerausgleich
Festsetzung der
Lohnsteuer

in 1985 gezahlte Steuern      DM   15800
fuer 1985 festgesetzt         DM    8972
verbleibende Betraege         DM    6828
Kirchensteuer gezahlt         DM    1650
Kirchensteuer festgestzt 9%   DM     807
verbleibende Betraege         DM     843
Rueckzahlung Sparzulage       DM     250
=====================================
Restguthaben                  DM    7421

  Berechnung
                        Ehemann    Ehefrau
Bruttoarbeitslohn DM      55000      15000
ab Freibetraege   DM       1080       1080
ab Werbungskosten DM       2263       1830
Einkuenfte        DM      51657      12090
=====================================

Anlage V          DM      -8000
Summe der Eink.   DM      43657      12090

  Berechnung          <Ende>

Gesamtbetrag der Einkuenfte   DM    55747
Kichensteuer (abziehbar)      DM     1340
Aussergewoehnl. Belastungen:
gezahlt        DM     2500
abziehbar                     DM        0
Versicherungsbeitraege:
gezahlt        DM     8000
abziebar                      DM     7074
=====================================
Zu versteuerndes Einkommen    DM    47333
```

Ob Sekt oder Selters, Rück- oder Nachzahlung – das Programm liefert es Ihnen schwarz auf weiß.

```
10 REM *********************************
20 REM *  lohn und einkommensteuer 1985*
30 REM *  (c) by michael hanke         *
60 REM *********************************
70 DIM a(44),b(24),c(39),d(10),e(10),f(24),y$(6),s(4)
75 REM on break gosub 4700
80 11=28
85 PAPER 0:PEN 1
90 BORDER 26:INK 0,26:INK 1,26:CLS:MODE 1
100 PRINT" L O H N - E I N K O M M E N S T E U E R"
110 PRINT:PRINT"                f u e r "
112 WINDOW #1,7,33,7,17
114 PAPER #1,13:CLS #1:PEN #1,0
116 PRINT #1
120 PRINT #1,"    1  99999   888   55555"
130 PRINT #1,"   11  9   9  8   8  5"
140 PRINT #1," 1 1  9   9  8   8  5"
150 PRINT #1," 1  1 9   9  8   8  5"
160 PRINT #1,"    1  99999   888   55555"
170 PRINT #1,"    1      9  8   8      5"
180 PRINT #1,"    1      9  8   8      5"
190 PRINT #1,"    1      9  8   8      5"
200 PRINT #1,"    1  99999   888   55555"
210 LOCATE 1,20:PRINT"(1) (- Berechnung der Jahressteuer 1985"
215 PRINT:PRINT"(2) (- Einkommensteuertabelle      1985"
216 PRINT:PRINT:PRINT"            ";CHR$(164);" M.Hanke 1985":BORDER 0:I
NK 0,0
220 a$=INKEY$:IF a$="1"THEN 230
222 IF a$="2"THEN 5000
225 GOTO 220
230 CLS:PRINT"        Hinweise zu den Eingaben !"
240 PRINT:PRINT:PRINT"Grundsaetzlich sind immer nur"
245 PRINT:PRINT"volle DM - Betraege einzugeben."
250 PRINT:PRINT"Die Eingaben sind mit der Enter-Taste"
255 PRINT:PRINT"abzuschliessen,Ausnahmen sind (j/n) Ab-"
260 PRINT:PRINT"fragen.Falscheingaben koennen durch"
265 PRINT:PRINT"DEL korrigiert werden."
270 PRINT:PRINT"0 DM Eingaben,oder nicht zutreffende"
275 PRINT:PRINT"Abfragen koennen durch Druecken der"
280 PRINT:PRINT"Enter-Taste beantwortet werden."
310 GOSUB 3370
320 s$="Steuerpflichtigen":e$="Ehefrau":r$="Religion":g$="Geburtsdatum"

330 k$="katholisch":v$="evangelisch":l$="ledig":h$="verheiratet"
340 az$="Angaben laut Zeilen Nr.":an$="Aus Anlage N fuer":ki$="Kirchenst
euer"
350 CLS:PAPER 13:PEN 0:PRINT" Allgemeine Angaben ":PAPER 0:PEN 13:a(1)=0
360 PRINT:PRINT:PRINT"Familienstand des "
370 PRINT:PRINT+s$+";  (v) = "+h$
380 PRINT"                 (l) = "+l$
390 y$=INKEY$:IF y$="" GOTO 390
400 IF y$="v"THEN LOCATE 21,6:PRINT"     "+h$+" ":GOTO 430
410 IF y$="l" THEN LOCATE 21,6:PRINT"    "+ l$+"          ":a(1)=1:GOTO 43
0
420 GOTO 390
430 PRINT"                                "
440 PRINT g$+" des        (TT.MM.JJ)"
450 PRINT:PRINT+s$+":       ##.##.##":y=10:GOSUB 3390:IF a(210102 THEN a
(2)=1
460 IF a(1)=1 GOTO 480
470 PRINT:PRINT+e$+":                ##.##.##":y=12:GOSUB 3390
475 IF a(210121 THEN b(2)=1
480 PRINT:PRINT r$+" des            ( J / E / K )"
490 PRINT:PRINT s$+":      ":GOSUB 3500:IF y$="k" THEN a(3)=2
500 IF y$="e" THEN a(3)=1
510 IF a(1)=1 GOTO 550
520 PRINT:PRINT e$;"          :       ":GOSUB 3500
530 IF y$="k"THEN b(3)=2
540 IF y$="e"THEN b(3)=1
550 PRINT:PRINT"Zahl der Kinder  :       ":
560 y$=INKEY$:IF y$=""THEN y$="*"
```

```
570 IF y$=CHR$(13)THEN y$="0"
580 IF ASC(y$)(48 OR ASC(y$))57 THEN 560
590 PRINT y$:a(4)=VAL (y$)
600 GOSUB 3540:IF a$="n" GOTO 350
610 CLS:PAPER 13:PEN 0:PRINT" Angaben zum Arbeitslohn ":PAPER 0:PEN 13
620 PRINT:PRINT an$+" den "+s$:b(0)=0:GOSUB 700:GOSUB 2390:IF a(1)=1 GOT
O 1670
630 CLS:PAPER 13:PEN 0:PRINT" Angaben in Anlage N fuer die "+e$+" ? ":PA
PER 0:PEN 13
640 PRINT:PRINT:PRINT:PRINT"              (j/n) ?":GOSUB 3550
650 IF a$="n"GOTO 1670
670 CLS:PAPER 10:PEN 0:PRINT" Angaben zum Arbeitslohn ":PAPER 0:PEN 13
680 PRINT:PRINT an$+" die "+e$:b(0)=1
690 GOSUB 700:GOSUB 2390:GOSUB 1670
700 GOSUB 850:GOSUB 1120
710 CLS:PAPER 13:PEN 0:PRINT" Werbungskosten allgemein ":PAPER 0:PEN 13
720 PRINT:PRINT:PRINT"Angaben in Zeile 45 - 48 ? (j/n)":GOSUB 3550

730 IF a$="j"THEN GOSUB 1400:GOTO 750
740 LOCATE 28,4:PRINT": keine":GOSUB 3540:IF a$="n" GOTO 710
750 CLS:PAPER 13:PEN 0:PRINT" Mehraufwendungen fuer Verpflegung ":PAPER
0:PEN 13
760 PRINT:PRINT:PRINT"Betraege in Zeile 49 - 54 ? (j/n)":GOSUB 3550
770 IF a$="j"THEN GOSUB 1480:GOTO 790
780 LOCATE 27,4:PRINT": keine":GOSUB 3540:IF a$="n"GOTO 750
790 CLS:PAPER 13:PEN 0:PRINT" Mehraufwendungen":PRINT:PRINT" fuer doppel
te Haushaltsfuehrung "
800 PAPER 0:PEN 13:PRINT:PRINT:PRINT"Angaben in Zeile 55 - 63 ? (j/n)":G
OSUB 3550
810 IF a$="j"THEN GOSUB 1570:GOTO 830
820 LOCATE 26,6:PRINT": keine":GOSUB 3540:IF a$="n" GOTO 790
830 f(22)=f(17)+f(19)+f(20)+f(21):IF f(22)(564 THEN f(22)=564
840 RETURN
850 :
860 PRINT:PRINT:PRINT"Bruttoarbeitslohn      DM  ######"
870 l=6:l1=28:GOSUB 3570:f(5)=y:GOSUB 3770
880 PRINT:PRINT"Lohnsteuer                   DM  ######":l=8:GOSUB 3570
890 f(6)=y:GOSUB 3770:x$=ki$
900 ON b(0)+1 GOTO 910,920
910 ON a(3)+1 GOTO 960,930,950
920 ON b(3)+1 GOTO 960,930,950
930 PRINT:PRINT x$+" (ev)     DM  ######":l=10:GOSUB 3570:f(7)=y:GOSUB 3
770
940 GOTO 960
950 PRINT:PRINT x$+" (rk)     DM  ######":l=10:GOSUB 3570:f(7)=y:GOSUB 3
770
960 LOCATE 1,12:PRINT"Betraege in Zeile 7 - 15 (j/n) ?":GOSUB 3550
970 IF a$="n"THEN LOCATE 24,12:PRINT":   keine":GOTO 1090
980 LOCATE 1,12:PRINT"Summe der Zeilen 7 - 9 DM  ######":l=12:GOSUB 3570

990 GOSUB 3770:f(8)=y
1020 PRINT:PRINT"Summe der Zeilen 10-12 DM  ######":l=14:GOSUB 3570:GOSU
B 3770:f(9)=y
1030 PRINT:PRINT"Summe der Zeile   13   DM  ######":l=16:GOSUB 3570:GOSU
B 3770:f(10)=y
1060 PRINT:PRINT"Summe der Zeilen 14+15 DM  ######":l=18:GOSUB 3570:GOSU
B 3770:f(12)=y
1090 GOSUB 3540:IF a$="n"THEN FOR i=5 TO 12:f(i)=0:NEXT:IF b(0)=1 GOTO 6
70 ELSE GOTO 610
1110 RETURN
1120 CLS:PAPER 13:PEN 0:PRINT" Werbungskosten ":PAPER 0:PEN 13:l=9
1130 PRINT:PRINT"Fahrten zw. Wohnung und Arbeitsstaette"
1140 PRINT:PRINT:PRINT"    PKW(1)  MOT(2)  MOF(3)  FAH(4)":GOSUB 3700:a=
VAL(a$)
1150 IF a=0 GOTO 1300
1160 ON a GOTO 1170,1180,1190,1200:GOTO  1260
1170 f(13)=0.36:w$="PKW":GOTO 1210
1180 f(13)=0.22:w$="MOT":GOTO 1210
1190 f(13)=0.16:w$="MOF":GOTO 1210
1200 f(13)=0.12:w$="FAH"
1210 PAPER 13:PEN 0:LOCATE 1,9:PRINT w$;:PAPER 0:PEN 13:PRINT" benutzt a
n Tagen       ######"
1220 GOSUB 3570:GOSUB 3770:f(14)=y
1230 IF f(14))365 THEN LOCATE 1,10:GOTO 1210
1240 PRINT:PRINT"Einfache Entfernung         ######":l=11:GOSUB 3570:GOSU
B 3770:f(15)=y
1250 IF f(15))999 THEN LOCATE 1,10:GOTO 1240
1255 GOSUB 3540:IF a$="n"THEN f(13)=0:f(14)=0:f(15)=0:GOTO 1120
1260 f(16)=f(13)*f(14)*f(15):f(17)=f(17)+f(16)
1265 LOCATE 1,25:PRINT"                                        "
1270 IF x(2 THEN LOCATE 1,14:PRINT"weitere Arbeitsstaetten  (j/n)?":GOSU
B 3550:ELSE GOTO 1310
1280 IF a$="j"THEN:x=x+1:GOTO 1120
1290 IF a$="n"THEN LOCATE 26,14:PRINT":  ";:PAPER 13:PEN 0:PRINT"keine":
PAPER 0:PEN 13
1300 GOSUB 3540:IF a$="n"THEN f(14)=0:f(15)=0:f(13)=0:x=0:GOTO 1120
1310 CLS:PAPER 13:PEN 0:PRINT" Werbungskosten ":PAPER 0:PEN 13
1320 PRINT:PRINT"Oeffentliche Verkehrsmittel"
1325 LOCATE 1,8:PRINT"Aufwendungen          : DM  ######":l=8:GOSUB 3570

1330 GOSUB 3770
1340 f(16)=y
1350 PRINT:PRINT:PRINT"vom Arbeitgeber"
1360 PRINT:PRINT"steuerfrei ersetzt   : DM  ######":l=13:GOSUB 3570:GOSU
B 3770
1370 f(18)=y
1380 GOSUB 3540:IF a$="n"GOTO 1310
1385 f(17)=f(17)+f(16)-f(18)
1390 RETURN
1400 x$="Betraege aus Zeile ":LOCATE 1,4:l1=30
1410 PRINT x$+" 45 :  DM ######":l=4:GOSUB 3570:GOSUB 3770:f(19)=y
1420 PRINT:PRINT x$+" 46 :  DM ######":l=6:GOSUB 3570:GOSUB 3770:f(19)=f
(19)+y
1430 PRINT:PRINT x$+" 47 :  DM ######":l=8:GOSUB 3570:GOSUB 3770:f(19)=f
(19)+y
1440 PRINT:PRINT x$+" 48 :  DM ######":l=10:GOSUB 3570:GOSUB 3770:f(19)=
f(19)+y
1450 PRINT:PRINT:PRINT"Summe in Feld 77       =  DM":y=f(19):l=13:GOSUB
3770
1460 GOSUB 3540:IF a$="n" GOTO 710
1470 RETURN
1480 x$="Betraege aus Zeile ":l1=30:LOCATE 1,4
1490 PRINT x$+" 50 :  DM ######":l=4:GOSUB 3570:GOSUB 3770:f(20)=y
1500 PRINT:PRINT x$+" 52 :  DM ######":l=6:GOSUB 3570:GOSUB 3770:f(20)=f
(20)+y
1510 PRINT:PRINT x$+" 53 :  DM ######":l=8:GOSUB 3570:GOSUB 3770:f(20)=f
(20)+y
1520 PRINT:PRINT"steuerfrei ersetzt     :  DM ######":l=10:GOSUB 3570:GO
SUB 3770
1530 f(22)=f(20)-y
1540 PRINT:PRINT:PRINT"Summe in Feld 76       =  DM":l=13:y=f(22):GOSUB
3770
1550 GOSUB 3540:IF a$="n" GOTO 750
1560 RETURN
1570 x$="Betraege aus Zeile ":l1=30:
1580 LOCATE 1,6:PRINT x$+" 58 :  DM ######":l=6:GOSUB 3570:GOSUB 3770:f(
21)=y
1590 PRINT:PRINT x$+" 59  :  DM ######":l=8:GOSUB 3570:GOSUB 3770:f(21)=f
(21)+y
1600 PRINT:PRINT x$+" 60  :  DM ######":l=10:GOSUB 3570:GOSUB 3770:f(21)=
f(21)+y
1610 PRINT:PRINT x$+" 61  :  DM ######":l=12:GOSUB 3570:GOSUB 3770:f(21)=
f(21)+y
1620 PRINT:PRINT"steuerfrei erstetzt     :  DM ######":l=14:GOSUB 3570:GO
SUB 3770
1630 f(21)=f(21)-y
1640 PRINT:PRINT:PRINT"Summe in Feld 75       = DM":l=17:y=f(21):GOSUB
3770
1650 GOSUB 3540:IF a$="n"GOTO 790
1660 RETURN
1670 CLS:PAPER 13:PEN 0:PRINT" Angaben aus Anlage V ":PAPER 0:PEN 13
1680 PRINT:PRINT"Eink. aus Vermietung und Verpachtung"
1685 LOCATE 16,6:PRINT"(j/n) ?":GOSUB 3550
1690 IF a$="n"THEN LOCATE 16,6:PRINT"keine  ":GOTO 1760
1700 LOCATE 1,6:PRINT"Setzen Sie die Summe ein,die Sie aus der"
1705 PRINT"Anlage V ermittelt haben."
1710 PRINT:PRINT"Bei negativen Einkuenften das - Zeichen "
1720 PRINT"vor der Eingabe nicht vergessen!"
1730 LOCATE 1,16:PRINT"Summe aus Anlage V :  DM ######":l1=26:l=16:GOSUB
 3570
1740 IF m=1 THEN y=y*-1
1750 GOSUB 3770:v=y
1760 GOSUB 3540:IF a$="n"THEN m=0:GOTO 1670
1770 CLS:PAPER 13:PEN 0:PRINT" Sonderausgaben ":PAPER 0:PEN 13
1775 x$="Sozialversicherung"
1780 PRINT:PRINT:PRINT x$+" des Steuerpflichtigen"
1790 PRINT"nur Arbeit";:PAPER 13:PEN 0:PRINT"nehmer";:PAPER 0:PEN 13:PRI
NT"anteil    : DM  ######"
1800 l1=32:l=6:GOSUB 3570:GOSUB 3770:a(23)=y:IF a(1) GOTO 1840
1810 PRINT:PRINT x$+" der Ehefrau"
1820 PRINT:PRINT"nur Arbeit";:PAPER 13:PEN 0:PRINT"nehmer";:PAPER 0:PEN
13:PRINT"anteil    : DM  ######"
1830 l=10:GOSUB 3570:GOSUB 3770:a(24)=y
1840 x$="Rentenversicherung"
1850 PRINT:PRINT:PRINT x$+" des Steuerpflichtigen"
1860 PRINT"nur Arbeit";:PAPER 13:PEN 0:PRINT"geber";:PAPER 0:PEN 13:PRIN
T"anteil    : DM  ######"
1865 l=15:IF a(1)=1 THEN l=11
1870 GOSUB 3570:GOSUB 3770:a(25)=y:IF a(1) GOTO 1910
1880 PRINT:PRINT x$+" der Ehefrau"
1890 PRINT:PRINT"nur Arbeit";:PAPER 13:PEN 0:PRINT"geber";:PAPER 0:PEN 1
3:PRINT"anteil    : DM  ######"
1900 l=19:GOSUB 3570:GOSUB 3770:a(26)=y
1910 GOSUB 3540:IF a$="n" GOTO 1770
1920 CLS:PAPER 13:PEN 0:PRINT" Sonderausgaben ":PAPER 0:PEN 13
1925 PRINT:PRINT"                    in 1985   in 1984"
1930 PRINT"                  gezahlt  erstattet:"
1940 PRINT:PRINT"Freiw. Vers.:  DM  ######":l1=20:l=6:GOSUB 3570:GOSUB 3
770
1950 a(27)=y
1960 LOCATE 28,6:PRINT"DM  ######":l1=32:GOSUB 3570:GOSUB 3770
1970 a(27)=a(27)-y
1980 PRINT:PRINT"Krankenvers.:  DM  ######":l1=20:l=8:GOSUB 3570:GOSUB 3
770
1990 a(28)=y
2000 LOCATE 28,8:PRINT"DM  ######":l1=32:GOSUB 3570:GOSUB 3770
2010 a(28)=a(28)-y
2020 PRINT:PRINT"Unfallvers. :  DM  ######":l1=20:l=10:GOSUB 3570:GOSUB
3770
2030 a(29)=y
2040 LOCATE 28,10:PRINT"DM  ######":l1=32:GOSUB 3570:GOSUB 3770
2050 a(29)=a(29)-y
2060 PRINT:PRINT"Lebensvers. :  DM  ######":l1=20:l=12:GOSUB 3570:GOSUB
3770
2070 a(30)=y
2080 LOCATE 28,12:PRINT"DM  ######":l1=32:GOSUB 3570:GOSUB 3770
2090 a(30)=a(30)-y
2100 PRINT:PRINT"Haftpflicht :  DM  ######":l1=20:l=14:GOSUB 3570:GOSUB
3770
2110 a(31)=y
2120 LOCATE 28,14:PRINT"DM  ######":l1=32:GOSUB 3570:GOSUB 3770
2130 a(31)=a(31)-y:GOSUB 3540:IF a$="n" GOTO 1920
2140 b(1)=a(27)+a(28)+a(29)+a(30)+a(31)
```

```
2150 CLS:PAPER 13:PEN 0:PRINT" Sonderausgaben ":PAPER 0:PEN 13
2155 PRINT:PRINT:PRINT"Bausparkasse Beitraege:  DM  ######":l1=30:l=4
2160 GOSUB 3570:GOSUB 3770:a(32)=y
2170 PRINT:PRINT"Kirchensteuer gezahlt :  DM  ######":l=6:GOSUB 3570:GOS
UB 3770:a(33)=y
2180 PRINT:PRINT"in 1984 erstattet     :  DM  ######":l=8:GOSUB 3570:GOS
UB 3770
2190 a(41)=y:a(33)=a(33)-a(41)
2200 PRINT:PRINT"Steuerberatungskosten :  DM  ######":l=10:GOSUB 3570:GO
SUB 3770:a(34)=y
2210 PRINT:PRINT"Berufsausbildung      :  DM  ######":l=12:GOSUB 3570:GO
SUB 3770:a(35)=y
2220 PRINT:PRINT"Spenden (Zeile 81-83) :  DM  ######":l=14:GOSUB 3570:GO
SUB 3770:a(36)=y
2240 PAPER 13:PEN 0:PRINT:PRINT" Aussergewoehnliche Belastungen ":PAPER
0:PEN 13
2242 PRINT:PRINT"Summe der Aufwendungen:  DM  ######":l=18:GOSUB 3570:GO
SUB 3770:a(43)=y
2244 GOSUB 3540:IF a$="n" GOTO 2150
2250 GOTO 2470
2260 IF a(0)=0 THEN e=e/2
2270 e=INT(e/54)*54
2280 IF e(=4265 THEN sx=0:GOTO 2360
2290 IF e)=18000 GOTO 2310
2300 sx=INT(0.22*e-926):GOTO 2360
2310 IF e)=60000 GOTO 2330
2320 a=(e-18000)/10000:sx=INT((((3.05*a-73.76)*a+695)*a+2200)*a+3034):GO
TO 2360
2330 IF e)=130000 THEN 2350
2340 a=(e-60000)/10000:sx=INT((((0.09*a-5.45)*a+88.13)*a+5040)*a+20018):
GOTO 2360
2350 sx=INT(0.56*e-14837)
2360 IF sx(0 THEN sx=0
2370 IF a(0)=0 THEN sx=sx*2
2380 RETURN
2390 FOR i=5 TO 22:GOSUB 2420
2400 NEXT
2410 RETURN
2420 IF b(0) GOTO 2440
2430 a(i)=f(i):f(i)=0:GOTO 2450
2440 b(i)=f(i):f(i)=0
2450 RETURN
2470 GOSUB 3860
2480 CLS:e$="":r$="":IF d=1 THEN e$=CHR$(1):r$=CHR$(18)
2485 x$=" Lohnsteuerausgleich ":y$="Lohnsteuer"
2490 IF f(1) THEN x$="Einkommensteuerveranlagung":y$="Einkommensteuer"
2500 :
2510 PAPER 13:PEN 0:PRINT #p,e$;r$;x$:PAPER 0:PEN 13
2520 PRINT #p:PRINT #p,e$;"Festsetzung der "+y$
2530 PRINT :PRINT #p:PRINT #p,"in 1985 gezahlte Steuern    DM ";:y=f(6):
GOSUB 3830
2540 PRINT:PRINT #p,"fuer 1985 festgesetzt       DM ";:y=et:GOSUB 3830
2550 PRINT:PRINT #p,"verbleibende Betraege       DM ";:y=f(6)-et:GOSUB 38
30
2560 IF a(7)+b(7)=0 AND e(5)=0 GOTO 2600
2570 PRINT:PRINT #p,"Kirchensteuer gezahlt       DM ";:y=a(7)+b(7):GOSUB
 3830
2580 PRINT:PRINT #p,"Kirchensteuer festgesetzt 9%DM ";:y=e(5):GOSUB 3830

2590 PRINT:PRINT #p,"verbleibende Betraege       DM ";:y=(a(7)+b(7))-e(5
):GOSUB 3830
2600 IF f(2)(=0 THEN f(2)=0:GOTO 2620
2610 PRINT:PRINT #p,"Rueckzahlung Sparzulage     DM ";:y=f(2):GOSUB 3830

2620 PRINT #p,"====================================="
2630 y= f(6)-et+a(7)+b(7)-e(5)-f(2)
2640 x$="Restguthaben":IF y(0 THEN x$="Nachzahlung "
2650 PAPER 13:PEN 0:PRINT:PRINT #p, x$;:PAPER 0:PEN 13:PRINT #p,"
      DM ";:GOSUB 3830
2660 IF p=8 THEN FOR a=0 TO 2:PRINT  #p:NEXT:p=0:GOTO 2700
2670 IF d=1 THEN p=8:GOTO 2485
2680 LOCATE 1,25:PAPER 13:PEN 0:PRINT"              (+)  weiter
     ";:PAPER 0:PEN 13
2690 a$=INKEY$:IF a$()";"GOTO 2690
2700 CLS
2705 PAPER 13:PEN 0:PRINT #p,e$;" Berechnung ":PAPER 0:PEN 13
2710 x$="Ehemann   Ehefrau":IF a(1) THEN x$="Steuerpflichtiger"
2720 :
2730 PRINT #p:PRINT #p,"                        "+x$
2740 PRINT:PRINT #p,"Bruttoarbeitslohn DM  ";:y=a(5):GOSUB 3800:IF a(1)T
HEN PRINT #p
2750 IF a(1)=0 THEN y=b(5):GOSUB 3830
2760 PRINT:PRINT #p,"ab Freibetraege   DM  ";:y=1080:GOSUB 3800:IF a(1)T
HEN PRINT #p
2770 IF a(1)=0 THEN y=1080:GOSUB 3830
2780 PRINT:PRINT #p,"ab Werbungskosten DM  ";:y=a(22):GOSUB 3800:IF a(1)
 THEN PRINT #p
2790 IF a(1)=0 THEN y=b(22):GOSUB 3830
2800 PRINT:PRINT #p,"Einkuenfte        DM  ";:y=a(5)-1080-a(22):IF y(0 T
HEN y=0
2810 GOSUB 3800:IF a(1)THEN PRINT #p:GOTO 2840
2820 y=b(5)-1080-b(22):IF y(0 THEN y=0
2830 GOSUB 3830
2840 PRINT:PRINT #p,"============================";:IF a(1)=0 THEN PRINT
 #p,"=========="
2845 IF v=0 GOTO 2870
2850 PRINT #p:PRINT #p,"Anlage V           DM  ";:y=v:GOSUB 3830
2870 PRINT:PRINT #p,"Summe der Eink.   DM  ";:y=c(2)+v:IF y(0 THEN y=0

2880 GOSUB 3800:IF a(1)THEN PRINT #p:GOTO 2910
2890 y=d(2):IF y(0 THEN y=0
2900 GOSUB 3830
2910 IF p=8 THEN FOR a=0 TO 2:PRINT #p:NEXT:p=0:GOTO 2960
2920 IF d=1 THEN p=8:GOTO 2705
2940 GOSUB 3740
2959 IF a$="-" GOTO 2480
2960 CLS
2965 PAPER 13:PEN 0:PRINT #p,e$;" Berechnung ";:PAPER 0:PEN 13:PRINT #p,
"  (Ende)"
2970 PRINT #p:PRINT #p,"Gesamtbetrag der Einkuenfte  DM  ";:y=c(3):IF y(
0 THEN y=0
2980 GOSUB 3830
2985 y=a(7)+b(7)-a(41):IF a(7)+b(7)=0 GOTO 2995
2990 PRINT:PRINT #p,"Kichensteuer (abziehbar)     DM  ";:GOSUB 3830
2995 y=a(42)+a(34)+a(35)+a(36):IF y=0 GOTO 3010
3000 PRINT:PRINT #p,"weitere Sonderausgaben       DM  ";:GOSUB 3830
3010 IF a(43)=0 GOTO 3080
3020 PRINT:PRINT #p,"Aussergewoehnl. Belastungen:"
3030 PRINT:PRINT #p,"gezahlt       DM  ";:y=a(43)
3040 GOSUB 3830
3050 PRINT:PRINT #p,"abziehbar                    DM  ";:y=a(44)
3060 :
3070 GOSUB 3830
3080 PRINT:PRINT #p,"Versicherungsbeitraege:"
3090 PRINT:PRINT #p,"gezahlt       DM  ";:y=c(6)
3100 GOSUB 3830
3110 PRINT:PRINT #p,"abziebar                     DM  ";:y=b(24)
3120 GOSUB 3830
3130 PRINT #p,"====================================="
3140 PRINT:PRINT #p,"Zu versteuerndes Einkommen   DM  ";:y=c(0):IF c(0)(
0 THEN y=0
3150 GOSUB 3830
3160 IF p=8 THEN PRINT #p,CHR$(29):p=0:d=0:GOTO 3190
3170 IF d=1 THEN p=8:GOTO 2965
3180 GOSUB 3740:IF a$="-" GOTO 2700
3190 CLS:PAPER 13:PEN 0:PRINT"     Lohn und Einkommensteuer 1985      ";
:PAPER 0:PEN 13
3200 LOCATE 8,10:PRINT"D (- Ausgabe auf Drucker"
3210 LOCATE 8,13:PRINT"N (- Neue Berechnung"
3220 a$=INKEY$
3230 IF a$=CHR$(13)THEN 2480
3240 IF a$="n"THEN CLS:CLEAR:GOTO 70
3250 IF a$="d"THEN d=1:PRINT #8,CHR$(1):GOTO 2480
3260 GOTO 3220
3370 a$=INKEY$:IF a$()" " GOTO 3370
3380 RETURN
3390 GOSUB 3480:FOR i =1 TO 6
3400 y$(i)=INKEY$:IF y$(i)=""THEN y$(i)="*"
3410 IF y$(i)=CHR$(127) GOTO 3490
3420 IF ASC(y$(i))(48 OR ASC(y$(i)))57 GOTO 3400
3430 LOCATE 26,y:PRINT y$(1)y$(2)"."y$(3)y$(4)"."y$(5)y$(6):NEXT
3440 a$=INKEY$:IF a$=CHR$(13) GOTO 3470
3450 IF a$=CHR$(127) GOTO 3490
3460 GOTO 3440
3470 a= VAL(y$(5)+y$(6)+y$(3)+y$(4)+y$(1)+y$(1)):RETURN
3480 FOR i=1 TO 6:y$(i)="#":NEXT:RETURN
3490 LOCATE 26,y:PRINT"##.##.##":GOTO 3390
3500 y$=INKEY$:IF y$="k" THEN PRINT k$:RETURN
3510 IF y$="e" THEN PRINT v$:RETURN
3520 IF y$="0" OR y$=CHR$(13) THEN PRINT"keine":RETURN
3530 GOTO 3500
3540 LOCATE 1,25:PAPER 13:PEN 0:PRINT"          alles richtig (j/n)
     ";
3545 PAPER 0:PEN 13:
3550 a$= INKEY$:IF a$="j" OR a$="n"THEN RETURN
3560 GOTO 3550
3570 GOSUB 3480:FOR i=1 TO 6
3580 y$(i)=INKEY$:IF y$(i)=""THEN y$(i)="*"
3590 IF y$(i)=CHR$(13) GOTO 3670
3600 IF y$(i)=CHR$(127)THEN GOSUB 3690:GOTO 3570
3610 m=0:IF i=1 AND y$(1)="-"THEN m=1:GOTO 3635
3620 IF ASC(y$(i))(48 OR ASC(y$(i)))57 GOTO 3580
3635 LOCATE 11,1:PRINT y$(1)y$(2)y$(3)y$(4)y$(5)y$(6):NEXT
3640 a$=INKEY$:IF a$=CHR$(13) GOTO 3670
3650 IF a$=CHR$(127)THEN GOSUB 3690:goto3570
3660 GOTO 3640
3670 y$=y$(1)+y$(2)+y$(3)+y$(4)+y$(5)+y$(6):y=VAL(y$)
3680 LOCATE 11,1:PRINT"       ":RETURN
3690 LOCATE 11,1:PRINT"######":RETURN
3700 a$=INKEY$:IF a$="" GOTO 3700
3710 IF a$=CHR$(13)THEN RETURN
3720 IF VAL(a$)=0 OR VAL(a$))4 GOTO 3700
3730 RETURN
3740 PAPER 13:PEN 0:LOCATE 1,25:PRINT"          (+) weiter  (-) zurueck
     ";:PAPER 0:PEN 13
3750 a$=INKEY$:IF a$="+"OR a$="-" THEN RETURN
3760 GOTO 3750
3770 LOCATE 11,1:PAPER 13:PEN 0:PRINT USING "######";y:PAPER 0:PEN 13:RE
TURN
3800 IF p()8 THEN PAPER 13:PEN 0:PRINT USING "######";y;:PAPER 0:PEN 13:
PRINT"    ";:RETURN
3810 PRINT #8,USING "######";y;:PRINT #8,"    ";:RETURN
3830 IF p()8 THEN PAPER 13:PEN 0:PRINT USING "######";y:PAPER 0:PEN 13:R
ETURN
3834 PRINT #8, USING "######";y:RETURN
3860 CLS:LOCATE 5,12:PRINT"Berechnung"
3870 c(1)=a(5)-600:c(2)=c(1)-480-a(22)
3880 IF a(1)=0 THEN d(1)=b(5)-600:d(2)=d(1)-480-b(22)
3890 IF d(2)(0 THEN d(2)=0
3900 IF c(2)(0 THEN c(2)=0
3910 c(3)=c(2)+d(2)+v:c(4)=a(33)+a(34)+a(35)+a(42):GOSUB 4550
3920 IF a(1)=1 THEN 3940
3930 IF c(4)(540 THEN c(4)=540
3940 IF c(4)(270 THEN c(4)=270
```

```
3950 c(5)=c(3)-c(4):c(7)=6000:IF a(1)=0 THEN c(7)=3000
3960 c(6)=a(23)+a(24)+a(27)+a(28)+a(29)+a(30)+a(31)
3970 c(8)=c(7)-a(25)-a(26):IF c(8)(0 THEN c(8)=0
3980 c(9)=c(6)-c(8):IF c(9)(0 THEN c(9)=0
3990 c(10)=c(6):IF c(8)(c(6)THEN c(10)=c(8)
4000 c(11)=c(9)+a(32):b(0)=2340+300*a(4)
4010 IF a(1)=0 THEN b(0)=4680+600*a(4)
4020 c(12)=b(0):IF c(11)(b(0)THEN c(12)=c(11)
4030 c(13)=c(11)/2-b(0)/2:c(14)=b(0)/2
4040 IF c(14))c(13)THEN c(14)=c(13)
4050 c(15)=c(10)+c(12)+c(14):c(16)=c(1):IF c(1))62400 THEN c(16)=62400

4060 b(4)=d(1):IF d(1))62400 THEN b(4)=62400
4070 c(16)=c(16)+b(4)
4080 c(17)=INT(c(16)*0.09):c(18)=c(17):IF c(17))b(0)THEN c(18)=b(0)
4090 c(19)=c(17):IF c(17))b(0)/2 THEN c(19)=b(0)/2
4100 b(23)=c(18)+c(19)
4110 c(20)=INT(b(23)/54+1)*54:IF a(1)=1 THEN 4130
4120 IF c(20)(600 THEN c(20)=600
4130 IF c(20)(300 THEN c(20)=300
4140 b(24)=c(15):IF c(15)(c(20)THEN b(24)=c(20)
4150 c(0)=c(5)-b(24):IF a(2)=1 THEN c(0)=c(0)-720
4160 IF b(2)=1 THEN c(0)=c(0)-720
4170 IF a(1)=1 AND a(4))0 THEN c(0)=c(0)-4212
4180 c(0)=c(0)-(a(4)*432)-a(44)
4190 IF a(1)=1 AND c(0))(24000+a(4)*900)THEN f(1)=1
4200 IF a(1)=0 AND c(0))(48000+a(4)*1800)THEN f(1)=1
4210 e=c(0)+a(12)+b(12):a(0)=a(1):GOSUB 2260
4220 et=sx*(c(0)/(c(0)+a(12)+b(12))):f(6)=a(6)+b(6)
4230 e(2)=a(10)+b(10):f(2)=e(2):IF e(2)=0 OR f(1)=1 THEN 4390
4240 IF a(9)624 THEN a(9)=624
4250 e(1)=23:IF a(4))2 THEN e(1)=33
4260 e(3)=INT(a(9)*e(1)+0.5)/100
4270 IF a(1)=1 THEN 4300
4280 IF b(9))624 THEN b(9)=624
4290 e(3)=INT(b(10)*e(1)+0.5)/100+e(3)
4300 IF a(8))624 THEN a(8)=624
4310 f(8)=a(8)-a(9):e(1)=16:IF a(4))2 THEN e(1)=26
4320 IF f(8)(0 THEN f(8)=0
4330 e(3)=INT(f(8)*e(1)+0.5)/100+e(3)
4340 IF a(1)=1 THEN 4380
4350 IF b(8))624 THEN b(8)=624
4360 f(8)=b(8)-b(9):e(3)=INT(b(8)*e(1)+0.5)/100+e(3)
4370 IF f(8)(0 THEN f(8)=0
4380 f(2)=e(2)-e(3):IF e(2)(0 THEN e(2)=0
4390 IF b(3)=0 AND a(3)=0 THEN 4550
4400 IF a(3)()0 AND b(3)()0 THEN e(4)=et:GOTO 4490
4410 IF a(3)=0 THEN 4460

4420 e=c(2):a(0)=1:GOSUB 2260
4430 e(4)=sx:IF a(1)=1 THEN e(4)=et:GOTO 4490
4440 e=d(2):GOSUB 2260
4450 e(4)=et*e(4)/(sx+e(4)):GOTO 4490
4460 e=c(2):a(0)=1:GOSUB 2260
4470 e(4)=sx:e=d(2):GOSUB 2260
4480 e(4)=et*sx/(sx+e(4))
4490 IF a(4)=0 THEN 4530
4500 IF a(4)=1 THEN e(4)=e(4)-600:GOTO 4530
4510 IF a(4)=2 THEN e(4)=e(4)-1500:GOTO 4530
4520 e(4)=e(4)-1560-((a(4)-2)*1800)
4530 IF e(4)(0 THEN e(4)=0
4540 e(5)=INT(e(4)*9)/100:IF e(5)(7.2 THEN e(5)=7.2
4550 e(8)=4:IF a(1)THEN e(8)=5
4560 IF a(1)=1 AND a(4))0 THEN e(8)=5
4570 IF c(3))100000 THEN e(8)=e(8)+1
4580 IF c(3))30000 THEN e(8)=e(8)+1
4590 IF a(4))0 AND a(4)(3 THEN e(8)=e(8)-2
4600 IF a(4))3 THEN e(8)=e(8)-2
4610 IF e(8)(=0 THEN e(8)=1
4620 e(7)=c(3)*e(8)/100:a(44)=a(43)-e(7):IF a(44)(0 THEN a(44)=0
4625 RETURN
4630 MODE 1:RETURN
4700 CLS:CLEAR:GOTO 70
5000 CLS:PAPER 13:PEN 0:PRINT"  Jahres - Einkommensteuertabelle 1985  ";
:11=34:l=4:PEN 13:PAPER 0
5010 PRINT:PRINT:PRINT"Zu versteuerndes Einkommen  : DM ######":GOSUB 35
70:GOSUB 3770
5015 ek=y:e=ek
5020 PRINT:PRINT:PRINT"Jahressteuer Grundtabelle   : DM "
5030 a(0)=1:GOSUB 2260:y=sx:s(1)=sx:l=7:GOSUB 3770
5040 PRINT:PRINT:PRINT"Jahressteuer Splittingtab.  : DM ":e=ek:a(0)=0:GO
SUB 2260:s(2)=sx
5050 y=sx:l=10:GOSUB 3770
5060 PAPER 13:PEN 0:PRINT:PRINT:PRINT"    Steuervorteil durch Abschreibu
ng    ";:PAPER 0:PEN 13
5070 PRINT:PRINT:PRINT"Abschreibungsbetrag         : DM ######":l=16:GOS
UB 3570:GOSUB 3770
5080 ab=y:ek=ek-ab:a(0)=1:e=ek:GOSUB 2260:s(3)=S(1)-sx
5090 PRINT:PRINT:PRINT"Steuervorteil Grundtabelle  : DM ":l=19:y=s(3):GO
SUB 3770
5110 a(0)=0:e=ek:GOSUB 2260:s(4)=s(2)-sx
5120 PRINT:PRINT:PRINT"Steuervorteil Splittingtab. : DM ":y=s(4):l=22:GO
SUB 3770
5130 PRINT:PRINT:PAPER 13:PEN 0:PRINT"       Neue Berechnung (j/n) ?
     ";:PAPER 0:PEN 13:GOSUB 3550
5140 IF a$="j" THEN 5000
5150 CLS:CLEAR:GOTO 70
```

c't

PUBLIC DOMAIN SOFTWARE aus den USA.
Wir haben unser Angebot erweitert!

SIG/M User Group (für CP/M),	jetzt 240 Disketten	je **DM 21,90**
NEU! PICO-Net UG	jetzt 34 Disketten	je **DM 21,90**
IBM-PC User Group	jetzt 454 Disketten	je **DM 28,90**
Ausführlicher Katalog der SIG/M oder PICO-Net, auf Diskette		je **DM 12,—**
Ausführlicher Katalog der IBM-PC UG (2 Disketten)		**DM 19,90**

Aus unserem Katalog: Software zum Superpreis, z. B.:

NEVADA-COBOL .. nur DM 99,—
UTAH-COBOL ..... DM 129,—

Für CP/M (NEVADA-COBOL) und MS-DOS (UTAH-COBOL). Der ideale Einstieg in diese mächtige Programmiersprache.
Eigenschaften:
- ca. 200 Schlüsselworte aus dem ANSI 74 Standard
- ausführliche Fehlermeldungen in englischer Sprache
- verlangt nur 48 KByte Hauptspeicher
- ausführliches Handbuch im Lieferumfang
- Lieferung ab Lager

Mit deutschem Handbuch!

SuperCopy ..... ab DM 299,—

Das Multiformatprogramm, mit dem Sie über 100 Formate für CP/M und MS-DOS lesen, schreiben und formatieren können! Übertragung auch von CP/M nach MS-DOS und umgekehrt! Für alle IBM-kompatiblen Rechner sowie für Alphatronic PC, P3 und ITT 3030.
- Formatliste wird ständig ergänzt.
- Auch deutsche Formate (MC-CP/M, NDR-Kleincomputer, TA P 50 usw.).
- Wir bieten Ihnen einen ständigen Updating-Service.
- Das eingestellte Format bleibt auch nach dem Warmstart erhalten.

Unser Spitzenangebot:
PMATE, Text-Editor .... DM 579,—

Textverarbeitung der Spitzenklasse!
Version 4.0 gerade erschienen! Der Texteditor, der sich allen Aufgaben anpaßt!
- Für CP/M 2.2, CP/M-86 und MS-DOS.
- Kommandos können zusammengefaßt werden zu sogenannten MAKROS. Bis zu 200 Makros gleichzeitig im Speicher.
- Echte Makrosprache.
- MS-DOS-Version läuft auch als Hintergrundprogramm. Maus wird unterstützt.
- Alle Kommandos können vom Benutzer umdefiniert werden. Sie schaffen sich so Ihr ganz persönliches Textsystem.

NEU! PSTAR-Makrosammlung für PMATE schafft eine Wordstar®-ähnliche Benutzeroberfläche, ohne die zusätzlichen Möglichkeiten von PMATE einzubüßen..... **nur DM 79,—**

SONDERAKTION bis zum Erscheinen der deutschen Version:
amerikanische Version mit ausführlichem Handbuch für nur ..... **DM 529,—**

Nutzen Sie unseren Kopierservice! Wir kopieren Daten und Programme von jedem Format auf jedes andere um. Auch von CP/M auf MS-DOS und umgekehrt! Z. Zt. ca. 200 Formate für CP/M und MS-DOS. Pro Diskette nur DM 39,90, incl. neuer Diskette, Versand und MwSt. Von und nach Apple CP/M DM 59,90.

Wenn Sie Fragen haben, rufen Sie uns an! Wir lassen Sie nicht alleine. Fordern Sie unseren kostenlosen Propekt an!

● **ComFood GmbH, Flaßkamp 24, 4400 Münster, Telefon 02 51/71 97 68** ●

– Wir füttern Ihren Computer –

ComFood Software GmbH

Des Schneiders Kern

Durchblick in der Innenwelt der CPCs

Teil 9: interrupts. . .ticker. . .fast ticker. . .frame flyback. . .events

Günther Woigk

Mit einem gewieften Interrupt-Konzept haben die CPC-Entwickler ein vielbeachtetes Novum auf dem Homecomputermarkt eingeführt. Nicht nur auf der Maschinensprachenebene, sondern auch in BASIC ist es damit möglich, mehrere Programme quasi gleichzeitig bearbeiten zu können. Man muß sich aber schon einige Mühe machen, um hinter die 'tickenden' und 'kickenden' Kulissen zu schauen.

Wie so oft, stützen sich die besonderen Fähigkeiten des Locomotive-BASIC sehr stark auf die Firmware. In diesem Fall sogar direkt auf den *Kernel*.

Das ist *die* Kontroll- und Schaltstation im Computer. Hier befinden sich alle Routinen, die zum Bank-Switching, für Hintergrund-ROMs, RSXes und ganz speziell zur Interrupt-Behandlung benötigt werden.

Aus eins mach viele, viele

Obwohl der *Kernel* fast beliebig viele Interrupt-Quellen simulieren kann, gibt es im Schneider CPC nur einen einzigen Timer-Interrupt: Alle 1/300 Sekunden erzeugt das Gate Array hierfür einen Impuls.

Die Z80-CPU wird im Interrupt-Modus 1 betrieben. Sie führt deshalb bei jedem Signal einen RST 7 aus. Hier steht jedoch nur ein Sprung zur Adresse &B939 (6er: &B941), da die eigentliche Interrupt-Routine im ungebankten RAM liegen muß.

Dort überschlagen sich dann die Ereignisse: Zunächst einmal werden die externen Interrupt-Quellen herausgefiltert, wie es bereits in der ersten Folge dieser Serie besprochen wurde. Danach wird ein Zeitzähler erhöht, der sich mit zwei Vektoren stellen und abfragen läßt (in BASIC mit der reservierten Variablen 'time'). Für reine Timing-Zwecke braucht man also keine Interrupts zu programmieren, die das System immer mit zusätzlicher Rechenzeit belasten. Hierfür reicht eine Abfrage des Zeitzählers aus.

Übrigens, wer die tatsächliche Uhrzeit samt Datum dauerhaft abrufbar zur Verfügung haben möchte, sollte sich in dieser Ausgabe mal den Artikel 'Die versteckte Uhr' anschauen.

Als letztes werden die Software-Interrupts abgearbeitet. Der *kernel* benutzt dazu vier Listen, in die das laufende Programm (im Normalfall also BASIC) eintragen kann, was wann zu tun ist. Einige Einträge werden vom Betriebssystem selbst bei seiner Initialisierung eingerichtet, so zum Beispiel Tastatur-Abfrage und Farb-Blinken.

Da gerade beim Interrupt-Mechanismus sehr viele spezielle, neuartige Begriffe auftauchen, sind diese im nebenstehenden Kasten zusammengefaßt und erklärt.

Vier Uhren

Der 'Interrupt Manager' kennt vier verschiedene Listen (*Chains*), die er völlig getrennt verwaltet.

Eine Liste, die *Fast Ticker Chain*, ist bei jedem Timer-Interrupt dran. Hier kann man Routinen eintragen, die ebenfalls 300mal pro Sekunde zu bearbeiten sind.

Während des Strahlrücklaufes des Monitors (wenn der Kathodenstrahl kein Bild zeichnen muß) wird die *Frame Flyback Chain* abgearbeitet, was beispielsweise für ein flimmerfreies SCROLL ausgenutzt wird.

Jeden dritten Interrupt (1/100 s) ist die *Sound Chain* dran und alle sechs Interrupts (1/50 s) die *Ticker Chain*, der Universal-Timer im CPC.

Die *Sound Chain* wird allerdings ausschließlich vom Sound Manager bedient. Nur die anderen drei stehen dem Hauptprogramm selbst zur Verfügung.

Hat man beispielsweise eine Routine, die 100mal pro Sekunde aufgerufen werden soll, kann man diese auf dem *Fast Ticker* 'einhängen' und muß dann noch eine kurze 'Geteilt-durch-Drei-Routine' vorschalten (300/3 = 100). Allerdings sollte man sich immer überlegen, ob nicht auch 50mal/s ausreichen. Dann kann man bereits den *Ticker* benutzen, der das

System nur mit einem Sechstel an Rechenzeit belastet.

Bei allen vorgesehenen Interrupt-Routinen (*Events*) ist zu bedenken, daß der *Kernel* sie bei jedem zugehörigen Interrupt aufs neue aufruft (so wie in BASIC das EVERY-Kommando). 'One Shots' (einmalige Unterbrechungen entspechend AFTER) bedürfen stets noch besonderer Vorkehrungen.

Am komfortabelsten ist der *Ticker*, mit dem sich auch am leichtesten solch ein 'One Shot' realisieren läßt. Bei ihm kann man mit einem Countdown-Wert angeben, nach wieviel 50stel Sekunden das *Event* ge*kick*t, also aktiviert werden soll, und einen Nachladewert, der die Wiederholzeiten festlegt. Ist dieser Nachladewert Null, so 'ruht' das *Event*. Will man also ein Äquivalent zum BASIC-AFTER programmieren, muß man als Nachladewert Null angeben. Leider wird der Block dabei nicht automatisch aus der *Ticker Chain* entfernt, sondern nur 'ruhiggestellt'. Er beansprucht deshalb auch weiterhin noch Rechenzeit, weil der *kernel* sich bei jedem *Ticker*-Interrupt an ihm 'vorbeihangeln' muß.

Ausgehend vom Timer-Interrupt der ULA sieht die Sache bisher also folgendermaßen aus: Der *Kernel* stellt anhand eines Zählregisters fest, welche Listen untersucht werden müssen und klappert darin alle Einträge, sogenannte *Block*s ab. Bei *Fast Ticker* und *Frame Flyback* tritt er die entsprechenden Routinen direkt an (*kicken*), auf daß sie irgendwann einmal ablaufen, beim Universal-*Ticker* erst, wenn der Countdown abgezählt ist.

Wann die Routinen aber tatsächlich aufgerufen werden, hängt von ihrer Art ab. Hier sind zunächst einmal *synchrone* und *asynchrone* Ereignisse zu unterschieden.

Asynchron

Die *asynchronen Events* werden gleich im Anschluß an die Interrupt-Behandlung gestartet. Da sie daher dem Hauptprogramm jederzeit dazwischenfunken können, dürfen sie ziemlich viele Unterprogramme der Firmware, zum Beispiel die der Text-VDU, nicht benutzen, weil diese nicht 'reentrant' sind. Nicht reentrant sind diejenigen Routinen, die unweigerlich 'ins Schleudern kommen', wenn sie ein Interrupt während ihrer Abarbeitung unterbricht und in der Interrupt-Routine erneut aufruft.

Expreß- (dringende) *asynchrone Events* werden sogar sofort aufgerufen, wenn der *kernel* feststellt, daß sie dran sind. Da in dieser Zeit weitere Interrupts verboten sind, dürfen solche Routinen nur sehr kurz sein und keine Restarts benutzen, da diese die Interrupts wieder zulassen würden.

Für *normale asynchrone Events* gelten die letzten beiden Einschränkungen nicht. Bei ihnen sind die Interrupts wieder zugelassen.

Will man aber in einer Interrupt-Routine Unterprogramme

Synchron

der Firmware benutzen (zum Beispiel Textausgabe für eine mitlaufende Uhr), muß man die Interrupts mit dem Hauptprogramm synchronisieren.

Synchrone Events werden vom *Kernel* deshalb nur in eine Warteschlange (*Pending Queue*) eingetragen und noch nicht gleich aufgerufen. Es ist Aufgabe des Hauptprogrammes, regelmäßig nachzuschauen, ob ein *synchrones Event* angestoßen ist und auf seine Ausführung wartet (*Polling*). Das sollte sinnvollerweise immer dann geschehen, wenn auch alle Routinen der Firmware benutzt werden dürfen.

Dieses *Polling* gestaltet sich leider relativ aufwendig. Statt dafür einen einzigen Vektor bereitzustellen, gibt es deren vier, von denen drei keine Alternative in ihrer Kombination lassen. Sie sind deshalb in Listing 1 noch einmal in ihrem Zusammenspiel dokumentiert:

Kernel	Kontroll- und Steuereinheit des Computers, Teil der Firmware
Interrupt	Unterbrechungssignal. Das laufende Programm wird kurz und unbemerkt unterbrochen, um eine spezielle *Interrupt*-Routine abzuarbeiten.
Chain	Eine Tabelle, die in einer besonderen Form im Speicher abgelegt wird: Die Listenelemente können an jeder beliebigen Stelle stehen und werden nur durch Zeiger verkettet.
Kettungs-Pointer	Zeiger zum Verketten der Listenelemente in einer *Chain*
Block	Listenelement in einer *Chain*
Event	Ereignis: Sozusagen ein Software-*Interrupt*. Dieser läßt sich vom laufenden Programm setzen, und sein Eintreten wird vom *Kernel* überwacht.
Kick,Kicken	Aus einem programmierten *Interrupt* wird ein *Event*, wenn der *Kernel* (eventuell auch das Hauptprogramm selbst oder eine External-*Interrupt*-Routine) das Ereignis aktiviert.
Synchroner Event	Dieses Ereignis wird nur ge*kick*t, aber erst auf Nachfrage durch das Hauptprogramm ausgeführt.
Polling	Nachfrage, ob ein *synchrones Event* ge*kick*t ist und abgearbeitet werden muß.
Pending Queue	*synchrone Event*s, die ge*kick*t wurden, werden vom *Kernel* in diese Warteschlange eingereiht, bis sie durch das Hauptprogramm abgearbeitet werden.
Priorität	Je nachdem, wie dringend ein *synchrones Event* ist, kann man ihm eine höhere *Priorität* zuordnen. Alle *Expreß-Event*s sind dringender als alle *normalen Events*. Die ge*kick*ten Ereignisse höherer *Priorität* werden in der *Pending Queue* immer vor *Event*s niedrigerer Dringlichkeit einsortiert, auch wenn diese schon in einem vorhergehenden *Interrupt* ge*kick*t wurden, so daß sie beim *Polling* zuerst behandelt werden. *Asynchrone Event*s haben keine zugeordnete *Priorität*.
Asynchrones Event **Expreß** **Normal**	Sobald der *Kernel* dieses Ereignis *kick*t, wird es auch sofort ausgeführt: während des *Interrupt*s! Wird dieses Ereignis ge*kick*t, wird es im Anschluß an den *Interrupt* ausgeführt, unabhängig vom Hauptprogramm, aber wieder mit erlaubten neuen *Interrupts* = > Restarts verwendbar
Fast Ticker Chain	Liste, in die Ereignisse eingereiht werden können, die mit jedem *Interrupt* ge*kick*t werden sollen (300mal/s).
Frame Flyback Chain	Liste für Ereignisse, die während des Strahlrücklaufes des Monitorbildes ge*kick*t werden sollen (nur *asynchron*, möglicherweise sogar nur *Expreß* sinnvoll).
Ticker Chain	Liste für Ereignisse, die alle 1/50 s oder seltener ge*kick*t werden sollen. Besonders flexibel, da man hier zusätzlich einen Countdown (in 1/50 s) vorgeben kann (Bildung von 'AFTER') und einen Restart-Wert (= > 'EVERY').
Einhängen	Übergibt man dem *Kernel* einen *Block*, damit er ihn in eine der Listen einträgt, spricht man vom *Einhängen*.
Aushängen	Der *Block* bleibt so lange in der Liste, bis er explizit ausgehängt, das heißt aus der Liste entfernt wird.
Arm	'Bewaffnen': Aktivieren eines *Event Block*s.
Disarm	'Entwaffnen': Ruhigstellen eines *Event Block*s.

Adr.	Bedeutung	Eingaben	Ausgaben	preserved: AFBCDEHLXY
&BCDA	Hänge *Block* in *Frame-Flyback*-Liste ein	HL zeigt	-/-	xx xx
&BCE3	Hänge *Block* in *Fast-Ticker*-Liste ein	auf den	-/-	xx xx
&BCE9	Hänge *Block* in *Ticker-Liste* ein	*Block*	-/-	xx xx
&BCDD	Hänge *Block* aus *Frame-Flyback*-Liste aus	HL zeigt	-/-	xx xx
&BCE6	Hänge *Block* aus *Fast-Ticker*-Liste aus	auf den	-/-	xx xx
&BCEC	Hänge *Block* aus der *Ticker*-Liste aus	*Block*	DE=Remain	xx xx
&BCF2	*Kick* einen *Event-Block* (siehe Text)	HL->*Block*		xx
&BCF5	Leere die *synchrone Event Pending Queue*	-/-	-/-	xxxx xx
&BCF8	*Disarm* einen *synchronen Event Block* und lösche evtl. Eintrag in *Pending Queue*	HL zeigt auf *Block*	-/-	xx
&BD0A	*Disarm* einen *asynchronen Event Block*	HL->*Block*	-/-	xxxxxxxx
&B921	Teste, ob ein *synchr. Event* wartet	-/-	CY =) ja	xxxxxxxx
&BCFB	Arbeite ein wartendes *synchr. Event* ab.	-/-	NC=) kein Ev.	xxxx xx
&BCFE	Diese 3 Routinen nur zusammenhängend	von &BCFB	-/-	xx
&BD01	benutzen (siehe Text)	von &BCFB	-/-	xx
&BD0D	Erfrage verbrauchte Zeit (in 1/300 s)	-/-	DEHL=Zeit	xxxx xx
&BD10	Setze Zeitzähler auf einen Startwert	DEHL=Zeit	-/-	xxxxxxxx
&BCB0	init. einen *Event Block* für einen Tonkanal	A=Kanalnr. HL->Block	-/-	xx

Die Vektoren zur Interrupt-Kontrolle

Die Routine ab Adresse #B921 testet, ob ein *synchrones Event* wartet. Da sie ganz im RAM liegt, werden hierfür noch keine Speicherbänke umgeschaltet. Es geht also recht flott, was auch notwendig ist, da man ja oft nachschauen muß, ob ein *synchroner Event* bearbeitet werden will. Wenn ja, liefert die Routine ein gesetztes Carry-Flag.

In diesem Fall muß man die Routinen KL NEXT-SYNC, KL DO-SYNC und KL DONE-SYNC in genau dieser Reihenfolge aufrufen. Das wird in dem Unterprogramm 'EXECUT' gemacht. Da KL NEXT-SYNC ebenfalls die *Pending Queue* testet, kann man 'EXECUT' so lange durchlaufen, bis hier das CY-Flag nicht mehr gesetzt ist.

Es wäre sogar sinnvoll, diese Routine direkt an die Routine KL NEXT-SYNC anzuschließen. Dann kann man sich das dreimalige Bank-Switching sparen, das mit dem Aufruf der drei Vektoren einhergeht, wenn man die zugehörigen Adressen erst einmal aus den Vektoren herausgeklaubt hat. Da die Vektoren KL DO- und -DONE - SYNC dann selbst nie mehr aufgerufen werden, könnte man sie für andere Funktionen vorsehen.

actio et reactio

Um einen Interrupt zu programmieren, muß man einen Eintrag in die gewünschte Liste vornehmen, anders ausgedrückt, einen Datenblock in die entsprechende *Chain* einhängen. Die einzelnen *Blocks* (hier mal englisch statt Blöcke) sind über *Kettungs-Pointer* miteinander verknüpft, so daß in jedem *Block* vermerkt ist, wo der nächste zu finden ist.

Kern eines jeden *Chain Block* ist der *Event Block*, dem man, einmal angestoßen, nicht mehr ansieht, woher der *Kick* kam.
Das 'Treten' kann dabei auf mindestens fünf verschiedene Arten erfolgen:

- Durch Einhängen eines Datenblocks in eine der drei Listen (*Ticker, Fast Ticker, Frame Flyback*)

- Oder indem man einen *Event Block* an den Vektor &BCB0 übergibt, um einen Interrupt zu programmieren, der eine *Sound Queue* nachlädt (wie ON SQ GOSUB).

- Oder man benutzt den Vektor &BCF2, der direkt ein Ereignis 'tritt': Das kann vom Hauptprogramm oder vom Interrupt-Pfad aus geschehen. Hiermit kann man sehr einfach mit einem *externen* Interrupt eine Behandlungs-Routine anstoßen und sich alle Möglichkeiten offenhalten, die der *Kernel* hierbei bietet. Einziger Wermutstropfen: Man kann nicht direkt über den Vektor gehen, da dieser mit einem Restart gebildet ist, der die – im Interrupt-Pfad verbotenen – Interrupts wieder zuläßt. Man muß sich also aus dem Vektor direkt die Adresse holen, was aber keine allzugroßen Schwierigkeiten bereitet (Listing 2).

In Tabelle 2 sind die benutzten Blöcke beschrieben: Im *Event Block* sind die Bytes 0 und 1 für den *Kettungs-Pointer* in der *Pending Queue* reserviert und dürfen unter keinen Umständen verändert werden!

Byte 2 zählt die eintreffenden *Kicks*. Einmal 'Treten' erhöht es, einmal Abarbeiten der Routine erniedrigt es. Damit ist sichergestellt, daß kein Aufruf ausgelassen wird, auch wenn das System mit dem Abarbeiten nicht nachkommt. Ist dieses Byte Null, steht kein *Kick* mehr zur Bearbeitung aus. Beim Initialisieren eines *Blocks* sollte das Byte deshalb immer auf Null gesetzt werden. Der Zähler geht allerdings nur bis 127. Negative Werte (-128. . .-1) bedeuten, daß dieses *Event* ruht, was zum Beispiel auch die *Event*-Routine selbst besorgen kann (um einen 'One Shot' zu simulieren).

Byte 3 enthält alle wesentlichen Informationen über die Art des Ereignisses: *Synchron, Expreß, Priorität* und ob eine FAR ADDRESS angegeben wurde.

Achtung! Im Firmware Manual im Appendix X sind bei der Beschreibung des *Event Blocks* die Bytes 2 und 3 vertauscht worden!

```
        ; Hauptprogramm
        ...
        CALL #B921          ; HI KL POLL SYNCHRONOUS
        CALL C,EXECUT       ; Falls etwas vorliegt
        ...
        ; weiter im Hauptprogramm

EXECUT: CALL #BCFB          ; KL NEXT SYNC
        RET  NC             ; Pending Queue leer
        PUSH AF
        PUSH HL
        CALL #BCFE          ; KL DO SYNC
        POP  HL
        POP  AF
        CALL #BD01          ; KL DONE SYNC
        JR   EXECUT         ; wiederholen bis leer
```

Listing 1: Das *Polling* der *synchronen Event Queue* läuft über insgesamt vier Vektoren.

Byte 4 und 5 sind die Adresse der *Event*-Routine, wobei Byte 6 eventuell noch das Bankselect-Byte für einen RST 3 angibt. Da der RST 3 Interrupts wieder zuläßt, ist eine FAR ADDRESS für *expreß-asynchrone Events* nicht zulässig!

Der Platz ab Byte 7 kann als Datenbereich für die *Event*-Routine genutzt werden, da diese hierfür eine Adresse im HL-Register übergeben bekommt. HL zeigt dabei auf Byte 5 (bei einer NEAR ADDRESS), sonst auf Byte 6 des *Event Block*s. (Ein ärgerlicher Schönheitsfehler!)

Die *Fast Ticker* und *Frame Flyback Chain Block*s bestehen außer aus dem *Event-Block* noch aus zwei weiteren Bytes für die *Kettungs-Pointer* in den jeweiligen *Chain*s.

Ebenso bei der *Ticker Chain*. Hier kommen aber noch zwei mal zwei Bytes für den Countdown und den Nachladewert hinzu.

Wie man's macht

Am einfachsten ist es, einen *Block* einzuhängen, der bis zum bitteren Ende in einer Liste verbleiben soll (zum Beispiel für eine Uhr). Über die ersten drei Vektoren aus Tabelle 1 wird ein *Chain Block* an den *Kernel* übergeben, und damit hat es sich.

Soll so ein Ereignis aber nur einmal stattfinden, wird es schwierig. *Expreß-asynchrone Events* können sich selbst überhaupt nicht aushängen, sondern höchstens durch Negativsetzen des Zählbytes im *Event Block* ruhigstellen. Bei der *Ticker Chain* kann das auch durch den Nachladewert Null erreicht werden. Dieser *Block* verbraucht dann aber nach wie vor Rechenzeit. Um ihn wieder ganz auszuhängen, muß man sich irgendeinen Trick einfallen lassen.

Alle anderen *Event*-Arten können die Vektoren &BCDD, &BCEG, &BCEC aufrufen, wodurch sie nicht mehr angestoßen werden. Es besteht dann allerdings die Gefahr, daß bereits weitere *Kick*s eingegangen sind, zum Beispiel (aber nicht nur) wegen zu langsamem *Polling*. Diese *Kick*s würden dann trotzdem noch abgearbeitet, was manchmal aber auch erwünscht sein kann.

Sonst muß man noch den Vektor &BCF8 beziehungsweise &BD0A aufrufen, mit denen die *Event Blocks* ruhiggestellt und, bei *synchronen Events*, auch aus der *Pending Queue* entfernt werden.

Im Hintergrund

Listing 3 zeigt eine mögliche Anwendung: einen Druckerspooler in seiner einfachsten Form.

Als Zwischenspeicher für die zu druckenden Zeichen dient ein Puffer von 256 Bytes (Zeile 120), der als 'Ringspeicher' organisiert wird. Dafür werden zwei Zeiger benötigt, die auf das erste im Puffer befindliche Zeichen und auf das letzte zeigen. Neue Zeichen werden immer hinten angefügt und die ältesten von vorne her 'weggedruckt'.

Kommt ein Zeiger am Pufferende an, macht er an dessen Anfang weiter (daher Ringpuffer), was hier ganz einfach dadurch erreicht wird, daß ein Byte nur bis 255 zählen kann. Der Puffer ist voll, wenn der hintere Zeiger (END) sich anschickt, den vorderen (ANF) zu überholen.

Mit Aufruf der Routine INIT (Zeile 50/220) wird zuerst der Vektor MC PRINT CHAR (im Listing: PRINT) auf die eigene Routine JOB umgebogen, die die zu druckenden Zeichen im Ringspeicher ablegt. Dann wird ein *Chain Block* auf dem *Fast Ticker* eingehängt, der die Routine EVROUT aufruft, die die Zeichen wieder aus dem Speicher ausliest und zum Drucker schickt.

Der *Chain Block* umfaßt die Zeilen 160 bis 200, der darin enthaltene *Event Block* beginnt in Zeile 170. Um vom *Polling* des BASIC-Interpreters unabhängig zu sein, wurde ein *asynchrones Event* gewählt, um aber andererseits Vektoren (und damit Restarts) in der *Event*-Routine verwenden zu können, 'nur' die normale Version. Da die Routine im zentralen RAM liegt, konnte auch auf eine FAR ADDRESS verzichtet werden (alles Zeile 190: Art-Byte).

Will ein Programm nun ein Zeichen auf dem Drucker ausgeben, wird von nun an JOB ab Zeile 290 abgearbeitet. JOB prüft zuerst, ob der Puffer nicht vielleicht voll ist. In diesem Fall kreist die CPU in den Zeilen 310 bis 350 und wartet darauf, daß, per Interrupt, ein Zeichen gedruckt wird.

Dann wird der Zeiger END erhöht und wieder abgespeichert. Um den entsprechenden Platz im Puffer zu finden, muß noch die Puffer-Basis addiert werden, was etwas umständlich ist, da hier ein Byte und ein Wort addiert werden müssen (Zeilen 380-420). Dann kann, in Zeile 440, endlich das Zeichen gespeichert werden.

Währenddessen wird nun, 300mal/s, EVROUT abgearbeitet. Diese Routine testet zuerst, ob überhaupt ein Zeichen auf seinen Ausdruck wartet (Zeilen 490-520). Danach, ob der Drucker bereit ist (Zeilen 530 & 540).

Nur wenn beide Tests erfolgreich verlaufen sind, wird das

```
EXTINT:EXX                  ; geht, da DI im Interrupt-Pfad
       LD   A,B             ; B' retten
       RES  2,B             ; Low-ROM-Bit
       OUT  (C),C           ; unteres ROM einschalten
       LD   B,A             ; B' restaurieren
       EXX                  ;
       LD   HL,(#BCF3)      ; Adresse aus Vektor holen
       RES  7,H             ; Bankselect-Angaben
       RES  6,H             ; zurücksetzen
       PUSH HL              ; indirekten Sprung vorbereiten
       LD   HL,EVENTB       ; Adresse des Event-Blocks
       RET                  ; Indirekter Sprung
```

Listing 2: Aufruf der *Kick*-Routine (&BCF2) vom Interrupt-Pfad aus.

```
+-------------------------------------------------------------+
|Event-Block: Byte 0,1: Kettungs-pointer für die Pending Queue |
|             Byte   2: Zähl- und Steuerbyte                   |
|             Byte   3: Art                                    |
|             Byte 4,5: Adresse der Behandlungsroutine         |
|     (evtl:) Byte   6: benötigte Speicherkonfiguration        |
|     (evtl:) Byte 7ff: Datenbereich für die Routine           |
|                                                              |
|zu Byte 3:   Bit  0  =0 =) 'near address' (=) Byte 6 unnötig) |
|                     =1 =) 'far  address' (via RST 3)         |
|             Bits 1-4      Event-Priorität (nur bei synchr.)  |
|             Bit  5     =) muß immer Null sein               |
|             Bit  6  =1 =) expreß, sonst normal               |
|             Bit  7  =1 =) asynchron, sonst synchron          |
+-------------------------------------------------------------+
|Frame Flyback Chain Block:                                    |
|             Byte 0,1: Kettungs-Pointer                       |
|             Byte 2ff: Event-Block                            |
+-------------------------------------------------------------+
|Fast Ticker Chain Block:                                      |
|             Byte 0,1: Kettungs-Pointer                       |
|             Byte 2ff: Event-Block                            |
+-------------------------------------------------------------+
|Ticker Chain Block:                                           |
|             Byte 0,1: Kettungs-Pointer                       |
|             Byte 2,3: Count-Down-Zähler bis zum Kick         |
|             Byte 4,5: Nachlade-Wert für den Zähler           |
|             Byte 6ff: Event-Block                            |
+-------------------------------------------------------------+
```

Jeder *Chain Block* besteht aus einem *Event Block*, einem *Kettungs-Pointer* und eventuell vier Zähler-Bytes.

Zeichen aus dem Puffer geholt, der Zeiger ANF weitergestellt (mit denselben Schwierigkeiten wie oben bei END) und das Zeichen via SEND gedruckt. Ganz wichtig ist hierbei, daß die Vektoren MC BUSY PRINTER (hier: BUSY) und MC SEND PRINTER (hier: SEND) nur benutzt werden konnten, weil es sich nicht um ein *Expreß-Event* handelt und weil auch nicht irgendwie an der Interrupt-Routine selbst herumgefummelt wurde!

Was diesem Beispiel noch zur vollen Einsatzreife fehlt, ist, daß der *Chain Block* ausgehängt wird, während der Puffer leer ist (weil der *Fast Ticker* doch ganz schön Zeit verbraucht). Optimal ist der Spooler aber erst, wenn man sich einen Hardware-Zusatz bastelt, der mit jedem Acknowledge des Druckers einen external Interrupt erzeugt, der dann, wie in Listing 2, den *Event Block kick*t. Dadurch würde erstens der Spooler schneller und zweitens das Gesamtsystem nicht mit der Rechenzeit für den *Fast Ticker Block* belastet.

```
; LISTING 3: DRUCKERSPOOLER
; ---------- --------------
;
        ORG  40000
        JP   INIT
;
ADFAST: EQU  #BCE3    ; Block einh.:Fast Ticker Chain
BUSY:   EQU  #BD2E    ; teste, ob printer busy (=> CY)
SEND:   EQU  #BD31    ; drucke Zeichen
PRINT:  EQU  #BD2B    ; warte auf not-busy
;                       und drucke Zeichen
BUFFER: DEFS 256      ; Ringspeicher (255 Zeichen)
ANF:    DEFB 0        ; Zeiger auf erstes Zeichen
END:    DEFB 0        ; Zeiger hinter letztes Zeichen
PATCH:  JP   JOB      ; Patch fuer WAIT
CHAINB: DEFW 0        ; Fast Ticker Block
EVENTB: DEFW 0        ; Event Block
        DEFB 0        ; Zaehlbyte
        DEFB %10000000 ; ART0 asynchron, normal,
                       ;near address
        DEFW EVROUT   ; Routinenadresse
;                     ; (ohne ROM-Select-Byte)
INIT:   LD   HL,PATCH; Vektor MC PRINT CHAR patchen:
        LD   DE,PRINT
        LD   BC,3
        LDIR
        LD   HL,CHAINB; FAST TICKER CHAIN BLOCK
        JP   ADFAST ; einhaengen & return
;
JOB:    PUSH HL
        PUSH AF
JOB1:   LD   HL,(ANF); L=ANF & H=END
        LD   A,H
        INC  A
        CP   L        ; ANF = END+1 ?
        JR   Z,JOB1   ; Speicher ist voll => warten
        LD   (END),A ; update END
        DEC  A
        LD   HL,BUFFER ; RINGSPEICHER-BASIS
        ADD  A,L      ; + ZEIGER
        LD   L,A
        JR   NC,JOB3
        INC  H
JOB3:   POP  AF
        LD   (HL),A   ; Zeichen speichern
        POP  HL
        SCF           ; Zeichen korrekt gedruckt
        RET           ; in CY anmerken
;
EVROUT:LD    HL,(ANF) ; L=ANF & H=END
       LD    A,L
       CP    H        ; ANF = END ?
       RET   Z        ; Leer
       CALL  BUSY
       RET   C        ; Drucker Busy
       INC   L
       LD    (ANF),HL ; Update
       LD    HL,BUFFER; Ringspeicher-Basis
       ADD   A,L      ; + Zeiger
       LD    L,A
       JR    NC,EV1
       INC   H
EV1:   LD    A,(HL)   ; Zeichen aus Speicher holen
       JP    SEND     ; Zeichen drucken & Ret
```

Listing 3: Ein Software-Spooler mit einem 'Ringpuffer' von 256 Bytes.

Neu:

Das große C64-Arbeitsbuch

Eine aktuelle und praktische Anleitung, das Optimale aus dem C64 herauszuholen – gültig auch für den C128.
Von Alexander **Pütz**.
544 Seiten, Großformat, 49 Abbildungen. Lwstr-gebunden. Ermäßigter Vorbestellpreis bis 30. 9. 86 **DM 78.–**, ab 1. 10. 86 kostet das Werk DM 95.–.
ISBN 3-7723-8121-9

Über das Buch:

Weit über hundert Programme, Routinen und Verfahren geben dem Praktiker in dem großen C64-Arbeitsbuch schlüssige Antworten auf die täglich an ihn herantretenden Fragen. Ganz gleich, ob Texte verarbeitet oder Dateien verwaltet werden sollen, Lehrprogramme zu erstellen sind oder die Werbung zu organisieren ist. Die ganze Fülle der Möglichkeiten, die sich mit Hilfe dieses Arbeitsbuches dem C64-Anwender auftun, verrät ein Blick auf das Inhaltsverzeichnis.
Der Basic-Programmierer benötigte bisher für seine Arbeiten eine ganze Reihe verschiedener Bücher und Unterlagen. Mit diesem großen C64-Arbeitsbuch wird sich das ändern. Jetzt hat er alles in einer Hand. Der Anwender kann mit Hilfe dieses Arbeitsbuches künftig beim Erstellen der Programme wesentlich besser zurechtkommen. Ja, er wird entdecken, daß der C64-Computer ihm einen großen Teil der ungeliebten und doch wichtigen und notwendigen Arbeiten abnimmt. Der umfangreiche Nachschlageteil erläutert dazu alle benötigten Basic- und DOS-Befehle kurz und bündig.
Jedem, der sich ernsthaft mit dem Basic-Programmieren beschäftigt, wird dieses große Arbeitsbuch bald eine unentbehrliche Arbeitshilfe sein. Der Anwender erfährt, daß Basic schließlich doch recht einfach ist und daß er sein Problem ohne fremde Hilfe selbständig lösen kann.

Franzis'
Franzis-Verlag, Postfach 37 01 20, München
(140)

HEISE / LUTHER

Postfach 61 04 07
3000 Hannover 61

Dieses Buch bietet eine große Auswahl an Finanzprogrammen, die in leicht verständlicher Form beschrieben sind. Sie können Ihren Computer die Zins-, Effektivzins-, Zinseszinsberechnungen nach dem amerikanischen u. europäischen Verfahren ausführen lassen; Börsen- und Aktienkurse verarbeiten, so daß Sie sofort sehen, wie sich Ihr Geld vermehrt.
Nr. 106 **DM 45,–**

Dieses Buch enthält acht BASIC-Programme aus den Fachbereichen Mathematik, Mechanik und Elektrotechnik. Es können z. B. Trafo- und Biegeträgerberechnungen sowie Torsions- u. Biegebelastung bei Motor und Getriebewellen berechnet werden.
Nr. 115 **DM 29,80**

Acht Programme für Commodore-Computer, die man nicht nur sofort einsetzen kann, sondern von denen man auch lernen kann, wie z. B. ein Logikanalysator nachgebildet wird.
Nr. 119 **DM 29,80**

Dieses Buch zeigt den schnellsten Weg von BASIC zu COMAL auf. Es bezieht sich auf den COMAL-Kernal, was bedeutet, daß es für alle COMAL-Versionen, so auch für die Version 0.14 bzw. 2.0 für Commodore-Computer oder Metanic-Comal für Apple gültig ist.
Nr. 108 **DM 36,80**

Eine Anleitung, wie man 50 bekannte C 64-Spiele erfolgreich meistert. Durch die guten Kurzbeschreibungen dient es auch als gute Orientierung vor der Anschaffung eines Spieles.
Nr. 48 **DM 29,80**

Dieses Buch ist ein Nachschlagewerk mit Demoprogrammen für die C 64 Erweiterungen Simon's Basic, Exbasic Level II, PASCAL 64, Logo und Forth. Es klärt jeden (!) Befehl und enthält zur Veranschaulichung viele Musterprogramme.
Nr. 124 **DM 36,80**

Sollten die Bücher nicht im Fachhandel erhältlich sein, bitte über Bestell-Coupon anfordern. Info-Katalog über das Luther-Gesamtprogramm kommt kostenlos mit.

Bestell-Coupon

Ja, senden Sie mir zu den ob. Preisen (zzgl. DM 3,50 Versandkostenpauschale) ☐ p.Nachnahme, ☐ Scheck anbei, folgende Bücher: (Best.-Nr. eintragen)

Name, Vorname

Straße

PLZ / Ort

Heise/ Luther

COMPUTER-LAUFWERKE

SHUGART

SA 455 5" slimline 2•40 Track IBM-PC	DM 395
SA 850/51 8" LW	DM 850/950
SA 604 5" HD 5 MB Winch.Platte 99ms	DM 795

TOSHIBA HD für IBM PC + kompatible

MK 53 FA 5" HD 45 MB 30ms	DM 5450
MK 54 FA 5" HD 63 MB 30ms	DM 6390
MK 56 FA 5" HD 86 MB 30ms	DM 7239

TANDON für IBM PC + kompatible

TM 100/2 5" DSDD, Lizenzbau Siemens	Preis auf Anfrage
TM 252 5" 10 MB HD Plattenlaufwerk, slimline	DM 1445
TM 262 20 MB HD slimline	DM 2050
TM 502 5" 10 MB HD Sonderpreis, volle Bauhöhe	DM 1130
TM 503 30 MB HD fullhight	DM 3900
Kontroler WD 1002 WX-2	DM 590

Ausführliche Preisliste kostenlos!

Preise incl. 14 % MwSt. + Porto und Verpackung.

ES GmbH, Taxetstr. 7, 8045 Ismaning
Tel. 0 89/96 75 72 od. 96 54 42, TX 5 213 786

dobert gmbh
Computer- und Sondertechnik
Virchowstraße 16
2000 Hamburg 50
Tel.: 040-389 49 24
389 49 25

DER UNIVERSELLE FLOPPY/HARD DISK CONTROLLER SASI/ECB BUS

- ★ Bis 4 Laufwerke
- ★ Floppy und/oder Hard Disks
- ★ Intelligent durch Single Chip Prozessor
- ★ Hoher Datendurchsatz durch vier DMA Kanäle
- ★ Hard Disk Interface: ST 412/506
- ★ Floppy Disk Interface: IBM/ECMA Format
- ★ SASI und ECB Anschluß
- ★ Beliebige Adreßlage
- ★ ECB Anschluß ohne Wait
- ★ **DM 948,— incl. MwSt.**

Bitte Info anfordern.

D/C

DER C-COMPILER ZUM SUPER-PREIS

NUR **DM 199.–**

- Erzeugt sehr schnellen und kompakten Code
- vollständige Version (ohne Long u. Float)
- Bibliothek (unix-kompatibel), über 100 Funktionen als Source
- deutsches Handbuch, 120 Seiten
- inkl. Assembler, linker und Screen-Editor
- sehr gut für Anfänger geeignet – über 50 Beispiel-Programme
- CP/M 80 oder MS-DOS (ohne Ass. u. Link)

WEITERE SPITZENANGEBOTE:

Microsoft C, V.3.0 (MS-DOS) ..... **DM 990.–**
C-Public-Domain-Disketten
C-Tools als Source

Versand per NN/VK, Ausland nur VK

DIPL.-ING. DECKER
Kalchreuther Straße 136, 8500 Nürnberg 10
Telefon 09 11/52 55 99

Reinhard Milde

Postfach 70 13 44
8000 München 70
Telefon 0 89/7 69 46 31

Alle Preise in DM für 1 Stück incl. 14 % MWSt zzgl. Versandspesen bei NN-Versand. Preise für größere Stückzahlen, OEM u. WV bitte anfragen! Angeb. freibleibend!

Diskettenlaufwerke

				ab 1	ab 3
EPS					
3,5"	SMD 120	0.5 MB		359,––	349,––
5,25"	SD 521	0.5 MB		359,––	349,––
5,25"	SD 540	1.0 MB		389,––	375,––
5,25"	SD 560	1.6 MB		410,––	398,––
5,25"	SD 580	1.0/1.6 MB		495,––	480,––
PAN					
3,5"	JU 363	1.0 MB–lp		450,––	438,––
5,25"	JU 475	AT-komp.		545,––	528,––

alle einschl. Datenb. + Stromversorgungsstecker

Datenbuch	10,––	Manual	35,––

Qume	142	0,5 MB		299,––	280,––
Shugart	SA 460	1,0 MB		339,––	320,––

Winchesterlaufwerke

NEC	D 5126	25.8 MB	1495,––	1450,––
Shugart	SA 712	12.9 MB	1095,––	1060,––
Rodime	RO 203	19.3 MB	1095,––	1060,––
SMS				
OMTI 5510		Controller	495,––	485,––
Archive		Streamer	60 MB	4200,––
Cipher		Floppy-Tape	20 MB	

Typenraddrucker

UCHIDA DWX 305	20 cps	699,––
JUKI 6100	22 cps	1498,––
JUKI 6200	30 cps	2680,––

Matrixdrucker

JUKI 5520, Farbe	180 cps	2190,––
OKI NML 182	120 cps	1098,––
OKI NML 192, NLQ	160 cps	1748,––
OKI NML 193, breit	160 cps	2248,––

FUJITSU DX 2100, DX 2200, DL 2400
NEU · SCHNELL · LEISE · FARBIG a. A.
Zubehör: Traktor, Farbbänder etc. a. A.

Halbleiterspeicher

	ab 1	ab 10	ab 25
4164–15 (Samsung)	3,70	3,65	3,55
4164–20	3,35	3,30	3,20
4164–15	4,20	4,10	3,95
4164–12	4,95	4,90	4,75
41256–15	7,50	7,30	7,––
41256–12	9,50	9,30	9,––
2716–35	8,90	8,50	8,10
2732–25	6,50	6,30	6,––
2764–25	6,50	6,35	6,20
27128–25	6,95	6,80	6,60
27256–20	13,20	13,––	12,80
27C64–25	9,95	9,80	9,60
6264–LP15	7,95	7,85	7,70

Integrierte Schaltungen

Z 80 und i805x			bitte anfragen
uPD 765	17,90	17,50	16,90
FDC 2793	43,50	41,75	39,95
FDC 9229 BT	69,––	67,––	a. A.

Disketten

		ab 10	ab 50	ab 100
FUJI				
3,5"	MF1DD	9,90	9,50	8,95
3,5"	MF2DD	13,50	12,90	12,25
5,25"	MD2HD	11,95	11,35	10,80
Wendedisketten				
5,25"	MD2D	2,95	2,80	2,60

Schnell, schneller, RAM-Disk

1 MByte Hochgeschwindigkeits-Massenspeicher für den ECB-Bus

Christian Kayser

Eine RAM-Disk ist eine feine Sache, und bei der KBit-Dichte und den Preisen der modernen RAM-ICs auch wirklich kein Luxus mehr. Man kann sie ziemlich leicht im Hauptspeicher installieren, kein Problem – wenn man genügend Speicher zur Verfügung hat. Sie nachträglich in ein bestehendes System zu integrieren, in einen Z80-ECB-Bus-Rechner zum Beispiel, erfordert dagegen oft erhebliche Eingriffe in Hard- und Software, vor denen mancher Hobbyist zurückscheut. Also bleibt die RAM-Disk am ECB-Bus ein Traum. . .? Hoffentlich nicht.

Einen Rechner um eine RAM-Disk zu erweitern, ist im Prinzip gar nicht so kompliziert, man braucht nur genügend Speicher. Und wenn der Rechner ein ECB-Bus-Rechner ist, dürfte es eigentlich überhaupt keine Schwierigkeiten geben, denn dieser Bus ist ja standardisiert. Speichererweiterung 'rein und fertig.

Dieser Idee möglichst nahe zu kommen, war das Ziel der Entwicklungsarbeit. Das gängige Konzept, die 'schnellste aller Scheiben' im direkten Zugriff der CPU anzusiedeln, also in einem Teil des Hauptspeichers, führt leider oft zu erheblichen Problemen. Zunächst einmal braucht man dafür genügend Hauptspeicher – ein paar hundert Kilobyte sollten es schon sein. Soviel kann 'Good old Z80' aus eigener Kraft aber gar nicht adressieren; eine Speicherverwaltungslogik, gleich welcher Ausführung, muß Hilfestellung leisten. Beim ECB-Bus äußert sich das normalerweise in zusätzlichen, über die Signale der CPU hinausgehenden Adressen.

Kam das zu erweiternde System bislang ohne eine derartige Schaltung aus, muß man ihm zusätzlich zur Speichererweiterung noch eine geeignete MMU (Memory Management Unit) 'einpflanzen'. Der Vorteil: Man kann sich zueinander passende Baugruppen aussuchen. Der Nachteil: Eine ziemlich teure Lösung.

Existiert bereits eine MMU, ist die Frage, ob die neuen Adreßsignale an den 'richtigen' Buspins liegen. Bedauerlicherweise wird der ECB-Bus bei diesen Signalen nicht mehr einheitlich gehandhabt [1]. Mitunter kann man dies bei der Wahl der RAM-Erweiterung berücksichtigen und so Arbeit sparen. Aber die nächste Hürde zum größeren Hauptspeicher kommt doch:

Eine Speichererweiterung im 'Stil der neuen Zeit' ist eine Europakarte voll 256-KBit-Chips, genau ein Megabyte. Viele MMUs erzeugen aber nur vier zusätzliche Adreßbits (A16. . .A19), gerade genug für 1 MByte Speicher. Eine RAM-Erweiterung *um* 1 MByte ist dann prinzipiell nicht mehr möglich, sondern nur *bis* zu dieser Größe. Da man die vorhandenen Systemspeicher ja schlecht weglassen kann, ergeben sich Überschneidungen, die es erst zu beseitigen gilt.

Diesen und anderen Problemen mit der Hardware kann man natürlich fast immer irgendwie abhelfen – mit Lötkolben, Messerchen und Schaltdraht biegt der echte Freak jedes System zurecht. Aber nicht jeder Besitzer eines ECB-Bus-CP/M-Rechners ist ein echter Freak.

Und der Zielvorstellung 'RAM-Karte 'rein und fertig' entspricht so viel Bastelei nun auch nicht gerade.

RAM Disk Controller

Bei der vorliegenden RAM-Disk-Karte wurde daher ein anderes Konzept verfolgt: Die Speicher der Disk sind für die CPU nicht mehr als 'Memory' zu erreichen, sondern als 'I/O'. Die Adreßaufbereitung für den 'Plattenspeicher' erfolgt völlig vom Systembus getrennt, dadurch können die genannten Schwierigkeiten mit mehr oder zuviel RAM gar nicht erst auftreten.

Um den Schaltungsaufwand zu reduzieren, ist die Adreßaufbereitungslogik ausschließlich dafür ausgelegt, ein Megabyte RAM unter CP/M als Laufwerk zu benutzen. Im Endeffekt verhält sich die RAM-Disk-Karte der CPU gegenüber fast wie ein gewöhnlicher Floppy-Controller einschließlich Laufwerk. Auch der 'RAM Disk Controller' besitzt ein 'Spur'- und ein 'Sektor'-Register, einen Daten- und einen Statusport. Auch hier werden die einzelnen Bytes eines 'Sektors' in aufeinanderfolgenden Zugriffen auf den Datenport gelesen beziehungsweise geschrieben. Die Speicherkapazität der 'Halbleiter-Platte' beträgt 256, 512, 768 oder 1024 Kilobyte, je nachdem, wie viele RAM-Bänke bestückt sind.

Für schnellstmögliche RAM-Disk-Operationen unter CP/M wurde die 'Sektorgröße' auf 128 Bytes festgelegt. Dies ist die einzige Sektorgröße, bei der ein CP/M-BIOS die Daten unmittelbar zwischen Port und Quell-/Zieladreßbereich umladen kann. (Bei größeren Sektoren müssen die Daten den Umweg über einen Blocking-/Deblocking-Puffer nehmen. Ein solcher Transfer in Etappen – von der Quelle in den Puffer und von dort zum Ziel – hat bei Floppy-Disks seinen Sinn, bei einer RAM-Disk dagegen verkompliziert und verlangsamt er nur den Zugriff.)

Durch das Controller-Konzept benötigt die RAM-Disk-Karte busseitig nur die zur Portadressierung absolut notwendigen Signale sowie ein Refresh-Signal gegen die 'dynamische Vergeßlichkeit' der Speicher-ICs. Bei diesen Signalen kann man sich dann auch wieder auf den ECB-Bus verlassen.

Die nötigen Software-Anpassungen beschränken sich bei einem CP/M-2.2- oder CP/M-Plus-BIOS meist auf die Disk-Parameter-Tabellen und die Transfer-Routinen. Letztere werden dadurch vereinfacht, daß alle vom BDOS gelieferten Werte (Spur, Sektor und DMA-Adresse) ohne Umrechnung übernommen werden können, außerdem gibt es beim 'RAM Disk Controller' keine Wartezustände wie bei Floppy-Controllern zu berücksichtigen. Die Initialisierung nach dem Einschalten (Löschen des Directory) kann vom BIOS aus erfolgen, aber auch durch ein geeignetes 'normales' Programm. Man muß diese Funktion also nicht unbedingt mit im BIOS unterbringen – zumindest nicht von vornherein. (Am Schluß dieses Artikels sind Beispieltabellen und -routinen für ein CP/M-2-BIOS abgedruckt. Mehr zu diesem Thema folgt in der nächsten c't.)

Universell

Die Schaltung der RAM-Disk ist so ausgelegt, daß sich die Karte nicht nur für Standard-ECB-Bus-Rechner mit Z80-CPU eignet. Sie arbeitet auch in Systemen mit einer anderen CPU, beispielsweise im c't180 (HD 64180) oder c't86 (8086). Die folgende Schaltungsbeschreibung wird die speziellen Gegebenheiten dieser Kombinationen berücksichtigen.

Funktional läßt sich die Schaltung in drei Teile gliedern:

- den Busanschluß (Portauswahllogik und Ports)
- die Speichersteuerung (Adreßmultiplexer, Refresh-Logik, $\overline{RAS}$- und $\overline{CAS}$-Erzeugung)
- den Speicher

An der Auswahl der Karte sind weder viele ICs noch viele Bussignale beteiligt. Da sich ein Portzugriff der Z80-CPU bereits an den Signalen $\overline{IORQ}$ und $\overline{M1}$ eindeutig erkennen läßt, kann die Hauptarbeit auf ein einziges IC 'abgewälzt' werden: den 8-Bit-Vergleicher 74LS688 (IC2, Vergleichstyp AMD 25LS2521). Sein Ausgang wird aktiv, wenn die Adreßsignale A2. . .A7 mit der Basisadresse der RAM-Disk übereinstimmen, $\overline{IORQ}$ aktiv und $\overline{M1}$ inaktiv ist. In dem Fall wertet ein Dekoder (IC8, LS139) auch A1 und A0 aus. Seine Ausgangssignale selektieren die einzelnen Ports.

Die Verriegelung mit $\overline{M1}$ verhindert, daß die Selektionslogik den Interrupt-Acknowledge-Zyklus der Z80-CPU mit einem 'richtigen' I/O-Zugriff verwechselt. Für die Hitachi-CPU HD 64180 des c't180 reicht dieser Schutz jedoch nicht aus. Denn so Z80-kompatibel dieser neue Prozessor auch ist, an einigen Stellen geht er eigene Wege, unter anderem ausgerechnet beim Timing für das I/O-Enable- und das Load-Instruction-Register-Signal (die entsprechenden Z80-Signale sind $\overline{IORQ}$ und $\overline{M1}$). Für den Einsatz an einem HD 64180 müssen deshalb statt $\overline{M1}$ die Signale $\overline{RD}$ und $\overline{WR}$ in die Verknüpfung einbezogen werden. Dazu dienen das Dioden-AND-Gatter (D2, D3 und R5) und die Lötbrücken J2 und J3 (per Layout für $\overline{M1}$ 'vorgejumpert').

Der c't86 hat ebenfalls ein spezielles Handikap: Wegen des 16-Bit-Datenbusses der 8086-CPU können 8-Bit-Ports nur auf geraden Adressen angesprochen werden. A0 ist also immer '0' und damit für die Auswahl der Ports ungeeignet. Mit den Steckbrücken J4 und J5 kann man jedoch A2 statt A0 an den Dekoder IC8 legen. (Die jeweils gültige Zuordnung der Ports zu den Adressen entnehmen Sie bitte nebenstehenden Tabellen.) Im übrigen gibt es beim c't86 kein $\overline{M1}$-Signal, allerdings auch keine 'Mißverständnisse' beim Interrupt-Acknowledge.

a		c
+ 5V	1	+ 5V
D5	2	D0
D6	3	D7
D3	4	D2
D4	5	A0
A2	6	A3
A4	7	A1
A5	8	A8
A6	9	A7
	10	
	11	IEI
	12	
	13	
	14	D1
	15	
	16	IEO
	17	
	18	
	19	
$\overline{M1}$ +	20	
	21	
	22	$\overline{WR}$
	23	
	24	+ $\overline{RD}$
	25	
	26	
$\overline{IORQ}$	27	
$\overline{RFSH}$ +	28	
	29	
	30	+ $\overline{MREQ}$
	31	
Gnd	32	Gnd

Mehr Bussignale braucht die RAM-Disk nicht (die markierten wahlweise).

Unidirektional

Drei der Ports – Statusport, Spur- und Sektorregister – arbeiten jeweils nur in eine Richtung: Die beiden Register sind nur zum Schreiben da, der Statusport nur zum Lesen. Zugriffe in Gegenrichtung dürfen nicht vorkommen, weil es beim Statusport dann 'Datenkollisionen' gibt (CPU-Treiber kontra Statusport) oder die Register undefinierte Werte übernehmen (offener Datenbus). Die Hardware unterbindet solche 'Fehlgriffe' nicht, darauf muß der Programmierer schon selbst achten.

Sektor- und Spurregister (IC4 und IC5, LS374) übernehmen ihre 'Ladung' mit der steigenden (hinteren) Flanke des jeweiligen Select-Signals von IC8. Ihren Ausgängen entspringen 13 der 20 notwendigen Disk-RAM-Adressen, die restlichen sieben liefert ein Zähler (IC3, LS393). Der Zähler wird bei einem Zugriff auf das Sektorregister (IC4) über einen Inverter aus IC9 (LS240) gelöscht und nach jedem Datentransfer um eins weitergezählt (mit der steigenden Flanke des Datenport-Select-Signals). So adressiert er bei aufeinanderfolgenden Datentransfers jedesmal ein anderes Byte im RAM, bis zu 128 Mal. Er bestimmt also die Sektorlänge.

'Status' ist der Zustand der Steckbrücken J7 und J8, als Port dient die eine Hälfte von IC6 (LS244). Die drei Brücken von J8 erscheinen als Datenbits D5. . .D7, sie sind ausschließlich dem Anwender vorbehalten. Hier kann man zum Beispiel die bestückte Kapazität kodieren, wenn man diese per Software abfragen können will. Man kann die J8 zugeordneten Eingänge auch für ganz andere Zwecke benutzen, die noch nicht einmal etwas mit der

RAM-Disk zu tun haben müssen. Der Phantasie sind kaum Grenzen gesetzt.

Die Brücke J7 dagegen, deren Zustand man im Datenbit D1 erfährt, kann 'den ganzen Verkehr aufhalten': Solange J7 geschlossen ist, erhält der Datenport (IC13, LS245) seine Steuersignale über die zweite Hälfte von IC6. Ist sie jedoch offen, läßt dieses IC nichts mehr durch. In dem Fall bleibt der Datenport 'zu', weil ihn kein Freigabesignal ('0' an IC13, Pin 19) mehr erreicht, außerdem wird der Schreibimpuls für die Speicher blockiert (Bussignal $\overline{WR}$).

Mit einem Schalter an J7 kann man die Sperre nach Belieben ein- und ausschalten. So kann man sich beziehungsweise den Inhalt der RAM-Disk vor unvermutet 'durchgehenden' Disk-Routinen schützen, zum Beispiel beim Austesten von BIOS-Änderungen.

Zugreifen

Zählt man die 20 Adressen einmal nicht mit, braucht die ganze Speichersteuerung nur drei Signale: das Datenport-Auswahlsignal von IC8 (Pin 4) sowie die Signale $\overline{WR}$ und $\overline{RFSH}$ vom Z80-Bus.

Sobald der Datenport ausgewählt ist (IC8, Pin 4 geht auf '0'), werden über zwei AND-Gatter (IC1, LS00) die $\overline{RAS}$-Eingänge aller Speicher-ICs aktiviert und eine Zugriffskontrol-LED eingeschaltet. Parallel dazu erhält der Datentreiber (IC13) sein Freigabesignal, sofern nicht die Brücke J7 offen ist. Über die mit Invertern aufgebaute Laufzeitkette (IC9, LS240) gelangt das Auswahlsignal rund 40 Nanosekunden später an die Umschaltpins der Adreßmultiplexer (IC7, IC10 und IC12, LS258) und nach weiteren rund 20 ns an den zweiten Dekoder aus IC8, der einer der vier Speicher-Bänke das $\overline{CAS}$-Signal beschert.

Um Schreib- und Leseoperationen zu unterscheiden, braucht im Prinzip nur das $\overline{WR}$-Signal vom Bus an die $\overline{WE}$-Eingänge der Disk-RAM-ICs gelegt zu werden. Bei Schreibzugriffen auf Ports aktiviert die Z80-CPU den $\overline{WR}$-Ausgang nämlich schon zu Beginn des Zyklus, etwa gleichzeitig mit $\overline{IORQ}$. Das Schreibsignal liegt also immer schon einige Zeit vor dem $\overline{CAS}$-Impuls an, die Speicher-ICs führen sogenannte 'Early Write'-Zyklen aus, bei denen die Datenausgänge hochohmig bleiben. Die Gefahr von 'Crashs' auf dem internen Datenbus (wegen der zusammengeschalteten Ein- und Ausgänge der RAM-ICs) besteht demzufolge nicht.

Leider verhält sich die neue Hitachi-CPU auch hier nicht ganz Z80-konform. Sie aktiviert $\overline{IOE}$ nämlich nicht mehr zusammen mit $\overline{WR}$, sondern schon einen halben Taktzyklus vorher. Dies ist der zweite, wichtigere Grund für das bereits genannte Dioden-AND-Gatter: Ohne die Verknüpfung mit $\overline{WR}$ kommt das Datenport-Auswahlsignal zu früh, als daß die Zeitbedingungen für das 'frühe Schreiben' der RAM-ICs noch sicher eingehalten werden könnten.

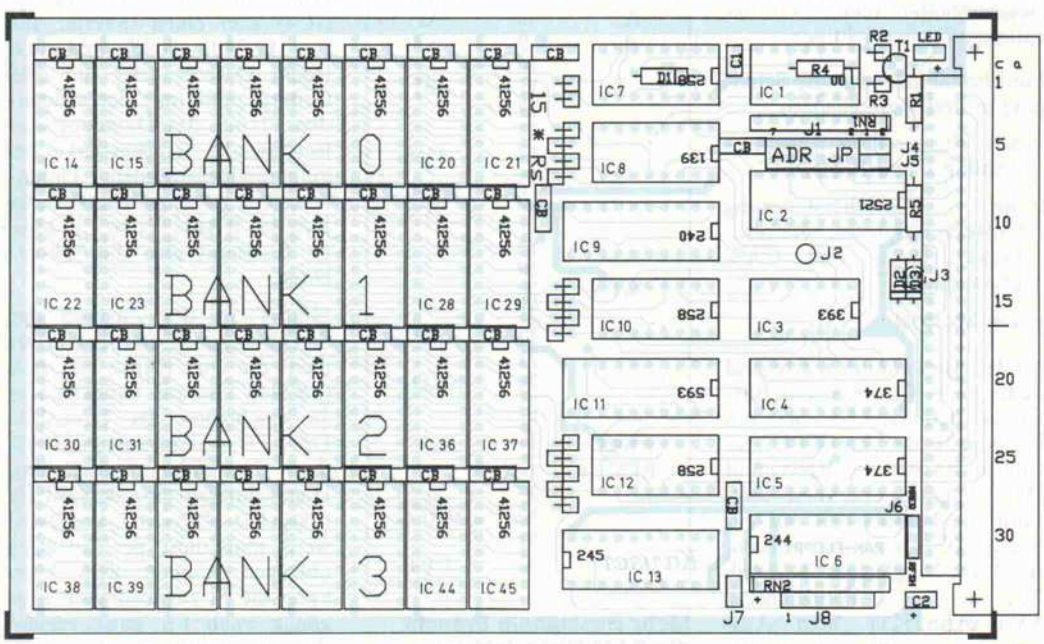
Erfrischt

Schließlich ist die Speichersteuerung noch für den Refresh der Speicher-ICs zuständig. Dazu wurde eins der 'exotischen' hochintegrierten LS-TTL-ICs eingesetzt (IC11): Der Typ LS593 beinhaltet einen 8-Bit-Zähler, der unter anderem Tri-State-Ausgänge aufweisen kann. Für den meistverwendeten 'RAS Only Refresh' braucht man außer diesem IC lediglich ein geeignetes Signal für den $\overline{RAS}$-Impuls.

Geeignet ist (fast) jedes Signal, das mindestens 256 Mal in 4 Millisekunden, aber nie gleichzeitig mit einem Zugriff auf den Datenport der RAM-Disk auftritt. Ein Refresh-Impuls muß mindestens die Zugriffszeit der RAM-ICs dauern (Richtwert) und genügend zeitlichen Abstand zum nächsten Impuls oder zu einem Disk-RAM-Zugriff halten. Diese Bedingungen erfüllt zum Beispiel das $\overline{RFSH}$-Signal der Z80-CPU.

Und so läuft ein Refresh-Zyklus ab: Wird $\overline{RFSH}$ aktiv, 'verabschieden' sich die Adreßmultiplexer ('1' an IC7, IC10, IC12, Pin 15), und der Refresh-Zähler 'meldet sich zu Wort' ('0' an IC11, Pin 18). Da der LS593 dafür relativ lange braucht, hält eine Verzögerungsschaltung den $\overline{RAS}$-Impuls ('0' an IC1, Pin 13) ein paar zig Nanosekunden auf. Am Ende des Zyklus – $\overline{RFSH}$ nimmt wieder Ruhepegel an – wird noch der Zähler inkrementiert (steigende Flanke an IC11, Pin 13).

Die 'lange Leitung' für den $\overline{RAS}$-Impuls ist ein einfaches



Beim Bestücken darf man R4 und D1 unter den Fassungen für IC1 und IC7 nicht vergessen.

Stückliste

Halbleiter

IC1	74LS00
IC2	74LS688 (25LS2521)
IC3	74LS393
IC4, IC5	74LS374
IC6	74LS244
IC7, IC10, IC12	74LS258
IC8	74LS139
IC9	74LS240
IC11	74LS593
IC13	74LS245
IC14-45	41256
T1	BC 558 *)
D1	1N4148
D2, D3	BAS 40, BAT 85, BAT 86
LED	3 mm *)

Widerstände

(1/4 oder 1/8 Watt)

R1	270R *)
R2	10k *)
R3	4k7 *)
R4	1k
R5	2k2
RN1, RN2	Array 8 x 4k7
RS	33R (15 Stück)

Kondensatoren

C1	150p keram.
C2	10µ/10V Ta.
CB	100n keram. (36 Stück)

IC-Fassungen

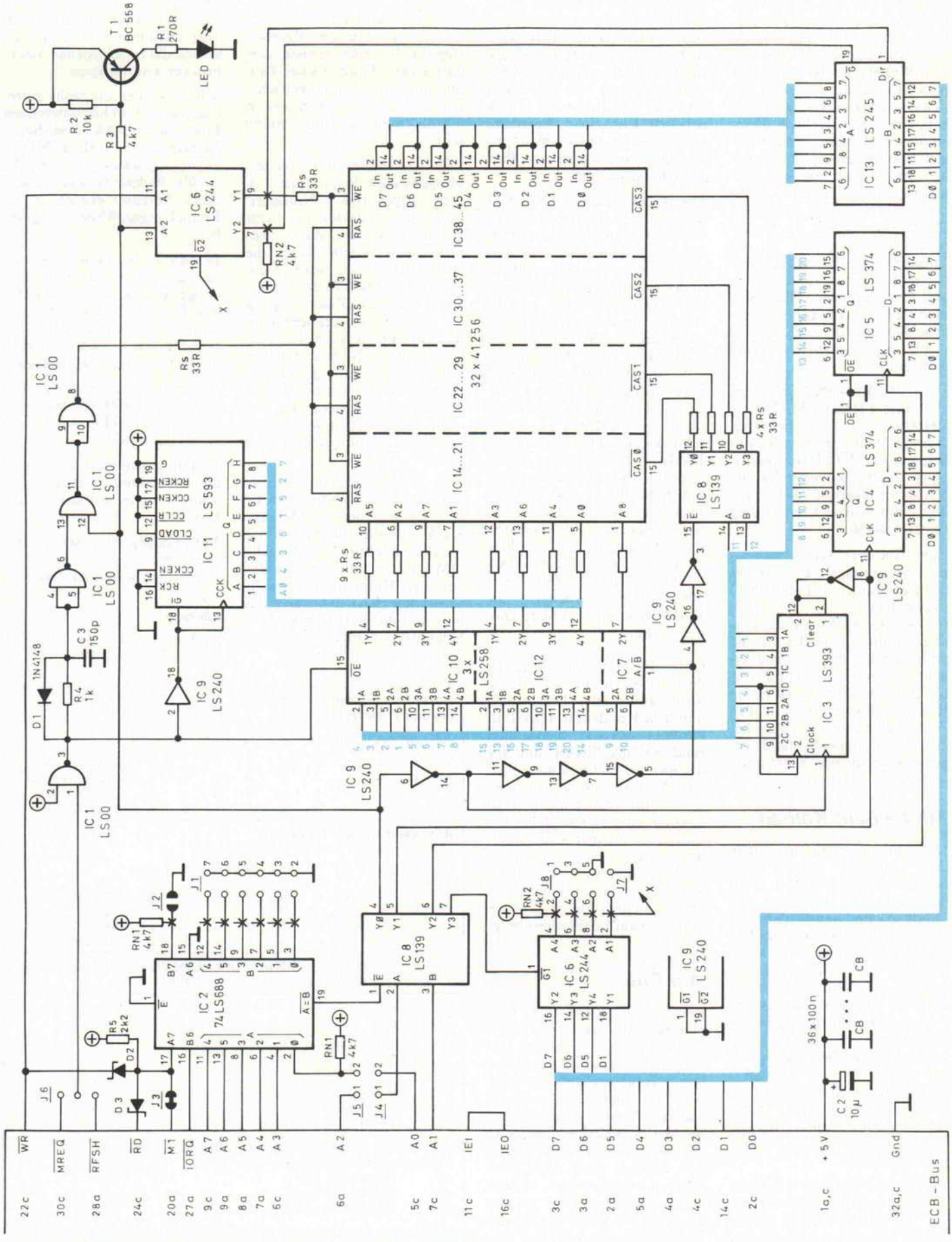
14polig	2 Stück
16polig	36 Stück
20polig	7 Stück

Sonstige Bauteile

J1, J4, J5	Pfostenleiste 2reihig, 16polig
J6	Pfostenleiste 1reihig, 3polig
J7	Pfostenleiste 1reihig, 2polig
J8	Pfostenleiste 1reihig, 6polig

Leiterplatte
64polige VG-Leiste
Steckbrücken.

*) Wenn man die Zugriffsanzeige nicht benötigt, können R1 ... 3, T1 und die LED entfallen.



Das Schaltbild der RAM-Disk-Karte sieht recht aufgeräumt aus.

RC-Glied (R4, C1), der wenigen Bauteile wegen. Damit sie möglichst nur die führende Flanke des RAS-Impulses verzögert, ist dem Widerstand die Diode D1 parallelgeschaltet. So wird der Zeitkondensator C1 schneller entladen. Der Nachteil dieser analogen Lösung, die vergleichsweise geringe Genauigkeit, ist hier von untergeordneter Bedeutung, das Bus-Timing der Z80-CPU läßt genügend 'Luft'. Hauptsache, die Verzögerungszeit ist größer als die Einschaltzeit der Zählerausgänge (maximal rund 50 ns). Mit der angegebenen Dimensionierung – dazu gehört ausdrücklich auch der LS00 als IC1 – gibt es normalerweise keine Probleme.

Für CPU-'Fremdgänger': Beim c't86 und c't180 ist MREQ als Refresh-Signal zu benutzen (Jumper J6). Ersterer stellt nämlich gar kein RFSH zur Verfügung, und bei letzterem hat die RAM-Ansteuerung wieder Probleme mit dem anderen Timing des HD 64180.

Zum Speicher selbst gibt es nicht viel zu sagen. Er ist in bewährter Manier als Feld von 32 dynamischen RAM-ICs (IC14. . .45, 41256) auf der busabgewandten Seite der Platine angeordnet. Jedes IC besitzt einen eigenen Abblockkondensator für die Stromversorgung (CB, 100 nF keramisch), in allen Adreß- und Steuerleitungen liegen Dämpfungswiderstände gegen hochfrequente 'Schmutzeffekte' (RS, 33 Ohm).

Mit heißem Kolben

Der Aufbau der Platine dürfte dem geübten Löter keine Schwierigkeiten bereiten, man darf nur nicht vergessen, die Diode D1 und den Widerstand R4 *vor* den Fassungen für IC1 und IC7 einzulöten. (Man sollte übrigens ruhig einmal probieren, ob der Widerstand auch wirklich unter die vorgesehene Fassung paßt. Bei manchen Billigsockeln wird es eng.) Ansonsten erfolgt die Bestückung wie gewohnt – erst die IC-Sockel und dann der Rest. Wenn man die Zugriffsanzeige nicht braucht, kann man R1 bis R3, den Transistor und die LED einfach weglassen.

Wer das erste Mal so viele dicht an dicht liegende Lötaugen wie bei den RAM-Sockeln hinter sich zu bringen hat, sollte darauf achten, daß sich die Platine nicht verzieht. Dies kann durch die starke örtliche Erwärmung beim Löten passieren, ist aber nicht weiter schlimm, solange die Leiterplatte nicht durch bereits verlötete IC-Fassungen 'krumm' gehalten wird. Zügig arbeiten kann man trotzdem, beispielsweise indem man pro Fassung nur ein oder zwei Pins auf einmal verlötet.

Die beiden Brücken J2 und J3 sind für den Betrieb an einer Z80-CPU voreingestellt (J2 offen, J3 geschlossen). Für den c't86 braucht man hier nichts zu ändern, es empfiehlt sich jedoch, den Widerstand R5 zu bestücken, da sonst der MI-Eingang des Vergleichers 'in der Luft hängt'. Bei der Hitachi-CPU des c't180 dagegen kommt man aus den genannten Gründen am Dioden-AND-Gatter nicht vorbei. Dazu ist J2 zu schließen und J3 zu öffnen. Die Dioden sollten Schottky-Typen sein, für die korrekte Funktion mit 'Feld-, Wald- und Wiesen'-Dioden (1N4148 und so weiter) wird keine Gewähr übernommen. Unumgänglich ist diese Änderung nur für die 64180-CPU, in einem Z80- oder 8086-System arbeitet die RAM-Disk mit und ohne Dioden-Gatter.

Ohne ICs und Steckbrücken darf die Karte nur sehr wenig Strom aufnehmen (Leckströme durch die Kondensatoren). Teilbestückt (nur TTL-ICs) sollten rund 300 mA fließen, und für die Speicher kommen noch etwa 50 mA dazu. Diese Angaben gelten ohne Steuersignale. Daß das eine oder andere IC im Betrieb etwas wärmer wird, vor allem der Refresh-Zähler (LS593), ist normal – aber nicht, wenn man sich die Finger verbrennt. In dem Fall müssen Sie sich doch auf Fehlersuche begeben.

Am Bus

Solange die RAM-Disk ordnungsgemäß 'refresht' wird, sind die hier gespeicherten Daten genauso sicher oder unsicher wie im Hauptspeicher des Rechners. Da dieser in aller Regel ebenfalls mit dynamischen RAM-ICs aufgebaut ist, sind die notwendigen Maßnahmen, den Refresh unter allen Umständen sicherzustellen, zumeist bereits getroffen. Dies gilt vor allem für das WAIT-Signal, das heutzutage ohnehin fast gar nicht mehr gebraucht wird, sowie DMA-Aktivitäten. In beiden Fällen wird die CPU 'lahmgelegt', so daß die Refresh-Impulse für eine gewisse Zeit ausbleiben. Trotz solcher Pausen muß ein vollständiger Speicher-Refresh (256 Impulse) in 4 Millisekunden durchlaufen sein.

Die Refresh-Impulse bleiben übrigens auch beim Reset aus. Der zugehörige 'Nottaster' sollte den Prozessor also nicht zu lange anhalten, damit man nicht nach jedem Reset die RAM-Disk neu laden muß. Wenn der Reset-Taster unmittelbar auf den Reset-Eingang der CPU wirkt, braucht man den berühmten 'schnellen Finger'. Sicherer ist eine dynamische Reset-Schaltung, die nur einen kurzen Impuls an die CPU abgibt (10 Mikrosekunden reichen völlig), auch wenn man den Finger auf dem Taster 'parkt'.

Von den 'Gänseblümchenketten' (Daisy Chains) des ECB-Bus ist auf der RAM-Disk-Karte nur die Interrupt-Kette durchgeschleift (IEI, IEO). Die DMA-Kette wurde aus Gründen der Kompatibilität nicht verdrahtet (BAI, BAO [1]), daher muß die RAM-Disk einen Steckplatz außerhalb der DMA-Kette erhalten.

Bevor Sie jetzt die Karte in Ihren Rechner stecken, um die ersten Tests zu fahren, sollten Sie noch einmal die verschiedenen Steckbrücken kontrollieren:

Mit J1 werden die sechs beziehungsweise fünf höchstwertigen Bits der RAM-Disk-Basisadresse gesetzt (offene Brücke = '1', geschlossene Brücke = '0'). Welche Brücke zu welchem Adreßbit gehört, ist im Bestückungsaufdruck angegeben.

J4 und J5 bestimmen, ob das Adreßbit A0 an den Portdekoder gelangt (beide gesteckt) oder das Bit A2 (Brücke von J4,1 nach J5,1), wie es der c't86 braucht. Im zweiten Fall muß die Brücke J1,2 offen bleiben.

J6 besorgt das passende Refresh-Signal (RFSH bei einer Z80-CPU, MREQ beim 8086 oder HD 64180).

J7 muß geschlossen sein, damit die RAM-Disk überhaupt beschrieben oder gelesen werden kann.

Mit J8 dagegen können Sie machen, was Sie wollen – die RAM-Disk kümmert's nicht.

Literatur

[1] Verbindend – der ECB-Bus, c't 12/85, Seite 60

Adressierung mit A0 und A1
(J4 und J5 gesteckt)

A7	A6	A5	A4	A3	A2	A1	A0	RD	WR	
x	x	x	x	x	x	0	0	j	j	Datenregister
x	x	x	x	x	x	0	1	n	j	'Sektor'register
x	x	x	x	x	x	1	0	n	j	'Spur'register
x	x	x	x	x	x	1	1	j	n	Statusregister

Adressierung mit A2 und A1
(Brücke von J4,1 nach J5,1; J1,2 offen)

A7	A6	A5	A4	A3	A2	A1	A0	RD	WR	
x	x	x	x	x	0	0	–	j	j	Datenregister
x	x	x	x	x	0	1	–	n	j	'Spur'register
x	x	x	x	x	1	0	–	n	j	'Sektor'register
x	x	x	x	x	1	1	–	j	n	Statusregister

Die 'x' repräsentieren die Basis-Portadresse der RAM-Disk, die mit J1 eingestellt wird (Jumper gesetzt – Adreßbit = '0').

Statusbyte

D7	D6	D5	D4	D3	D2	D1	D0
u2	u1	u0	–	–	–	S	–

S führt den Pegel des Schutz-Eingangs (J7):
S = '0' – RAM-Disk bereit
S = '1' – RAM-Disk geschützt

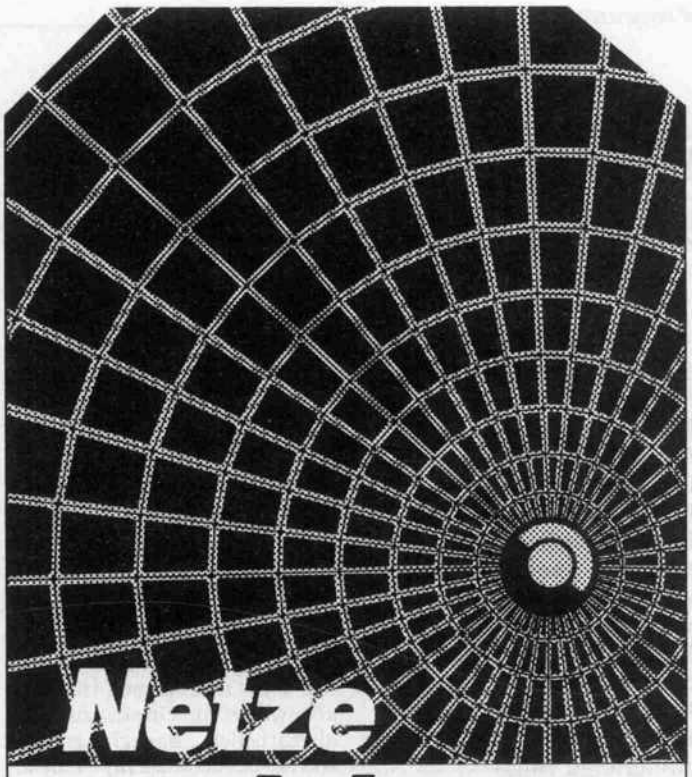
u0 . . . 2 sind die Steckbrücken von J8 und stehen zur freien Verfügung (Jumper gesetzt – Bit = '0'):
u0 – Kontakte 5 und 6
u1 – Kontakte 3 und 4
u2 – Kontakte 1 und 2

Die übrigen Datenbits sind nicht beschaltet und liefern daher beim Lesen des Statusports kein verläßliches Ergebnis.

```
RD_DPH: DEFW    0       ;XLT=0, KEIN SKEW
        DEFW    0
        DEFW    0
        DEFW    0       ;6 BYTES SCRATCH
        DEFW    DIRPUF  ;IM BIOS DEFINIERT
        DEFW    RD_DPB
        DEFW    0       ;CSV=0, WEIL CKS=0 (DBP)
        DEFW    RD_ALV
;
RD_DPB: DEFW    32      ;SPT=32 FÜR BANKE 0...3
                        ;   =24 FÜR BANKE 0...2
                        ;   =16 FÜR BANKE 0,1
                        ;   = 8 NUR BANK  0
        DEFB    4       ;BSH,
        DEFB    15      ;BLM FÜR 2K-BLöCKE
        DEFB    0       ;EXM=0  FÜR DSM>255
                        ;   =1  FÜR DSM<256
        DEFW    511     ;DSM=511 FÜR BANKE 0...3
                        ;   =255 FÜR BANKE 0...2
                        ;   =127 FÜR BANKE 0,1
                        ;   = 63 NUR BANK  0
        DEFW    191     ;DRM FÜR 192 DIR.ENTRIES
        DEFB    11100000B ;AL0,
        DEFB    00000000B ;AL1 FÜR 3 DIR.BLöCKE
        DEFW    0       ;CKS=0 FÜR FESTES MEDIUM
        DEFW    0       ;OFF=0, KEINE SYS.SPUREN
;
RD_ALV: DEFS    64      ;ALLOCATION VECTOR FÜR
                        ;DSM=511 (ALLGEMEIN:
                        ;ALV: DEFS (DSM/8)+1)
;-----------------------------------------------

RD_BASIS EQU    08H     ;RAM-DISK-BASISADRESSE
                        ; (BEISPIEL)
RD_DATEN EQU    RD_BASIS
RD_SEKT  EQU    RD_BASIS+1
RD_SPUR  EQU    RD_BASIS+2
RD_STAT  EQU    RD_BASIS+3


RD_READ: CALL   SET_RDC         ;SETZE REGISTER
         RET    NZ              ;-) BDOS-ERROR,
                                ;WENN GESCHÜTZT
         INIR                   ;BLOCK-TRANSFER
         RET

RD_WRITE:CALL   SET_RDC         ;SETZE REGISTER
         RET    NZ              ;-) BDOS-ERROR,
                                ;WENN GESCHÜTZT
         OTIR                   ;BLOCK-TRANSFER
         RET

SET_RDC: IN     A,(RD_STAT)     ;RAM-DISK
         BIT    1,A             ;GESCHÜTZT?
         JR     NZ,RD_SCHUTZ    ;JA, FEHLER-
                                ;MELDG. +RETRY
         LD     A,(SPUR)
         OUT    (RD_SPUR),A     ;"SPUR", "SEKTOR"
         LD     A,(SEKTOR)      ;UND "DMA" IM
         OUT    (RD_SEKT),A     ;BIOS DEFINIERT
         LD     HL,(DMA)
         LD     C,RD_DATEN
         LD     B,128           ;FÜR INIR/OTIR
         XOR    A               ;A =0, Z-FLAG =1
         RET                    ;GEGEN BDOS-ERROR
;
RD_SCHUTZ:LD    HL,RD_SCH_ML    ;BIOS-EIGENE
         CALL   STRING_OUT      ;STRING-AUSGABE:
                                ;HL=STRING-ANF.
                                ;DELIMITER =0
         CALL   CON_IN          ;BIOS-KONSOL-
         CP     3               ;EINGABE
         JR     NZ,SET_RDC      ;RETRY, AUSSER
                                ;BEI CNTL-C:
         LD     A,1             ;KENNUNG FÜR
         OR     A               ;BDOS-ERROR
         RET
;
RD_SCH_ML:DEFB 0DH,0AH,'RAM-Disk geschuetzt. ^C'
          DEFB ' -) BDOS-Fehler; sonst Retry ',0
```

Die RAM-Disk-Parametertabellen für CP/M 2.2 und 'Low Level'-Schreib-/Leseroutinen mit Abfrage der Schutzfunktion.

ct



von Adcomp

mit dem Netzwerkstandard von Novell:

NetWare

PCnet und Omninet für kleinere Netze, G-net, Arcnet und Ethernet für mittlere Netze, Pronet für große Netzwerke bis zu 50 Stationen.

Netze mit Disksystemen bis 600 MB, mit Datensicherung, mit Wechselplatten, mit Verbindung zu Mainframes.

Netze, die fehlersicher sind, die auf 80286 Systemen die größeren Speicher nutzen und die kompatibel sind zu allen gängigen Datenbanksystemen.

Netze, die IBM-Terminal oder Telex spielen. Netze, die der mittleren Datentechnik Konkurrenz machen.

Netze, die über Coaxialkabel, verdrillte Leitungen oder Fiberoptik kommunizieren.

Und preiswerte Netze für limitierte Benutzeranzahl.

ADCOMP Datensysteme GmbH
Olgastraße 15, 8000 München 19
Telefon 089/129 80 45, Telex 5 216 271

BÜRO NORD-WEST:
Alter Markt 24–25,
4050 Mönchengladbach 1
Telefon 0 21 61/3 22 17

BÜRO BERLIN:
Ratiborstraße 3, 1000 Berlin 36
Telefon 030/618 88 02

Hannover Messe '86
CeBIT —— 12.-19. 3. 1986
Wir stellen aus:
Halle 3, Stand 50

adcomp

Schritt für Schritt

Atari ST im Single-Step-Modus

Ein unentbehrliches Hilfsmittel zum Austesten von Assembler-Programmen ist die kontrollierte Ausführung in Einzelschritten. Natürlich genügt es nicht, wenn der Prozessor nach jeder Instruktion 'stoppt'. Er muß vielmehr aus dem Einzelbetrieb in den Normalbetrieb versetzt werden und in ein geeignetes Hilfsprogramm verzweigen, damit der Programmierer sich mit CPU-Unterstützung Status und Registerinhalte anschauen kann. Ein solches Hilfsprogramm stellen wir in diesem Beitrag vor.

Der Einzelschritt-Modus als Testhilfe stand dem Programmierer bei Computern aller Generationen zur Verfügung. Bei älteren Mini-Computern, beispielsweise DEC PDP 9, mußte er eine Einzelschritt-Taste am Kontroll-Panel betätigen. Bei den modernen Prozessoren hat man die Steuerung des Einzelschrittbetriebs aber sehr elegant von der Hardware- auf die Software-Ebene verlagert. Kennzeichen dieser Neuerung ist beim 68000-Prozessor des Atari ST ein Trace-Bit (T-Bit) im Statusregister.

Wie funktioniert der Einzelschrittbetrieb? – Wenn das T-Bit gesetzt ist, führt der Prozessor jeweils nach Ausführung einer Instruktion einen sogenannten Trace-Trap durch. Er kopiert Programmzähler und Statusregister auf den System-Stack und lädt den Programmzähler mit dem Trace-Vektor. Beim 68000 steht dieser an der Adresse $24.

Wichtig ist dabei: Durch den Trap wird das Trace-Bit im Statusregister gelöscht; gleichzeitig wird das System-Bit gesetzt. Die zugehörige Trap-Routine wird infolgedessen im Normalbetrieb auf der System-Ebene abgearbeitet. In dieser Routine kann man nun die erforderlichen Tests durchführen.

Abgeschlossen wird die Trap-Routine mit der Ausführung der RTE-Instruktion. Dieser Befehl lädt den geretteten Programmzähler und das Statusregister wieder zurück. Da in der Kopie des alten Statusregisterinhalts das T-Bit noch gesetzt ist, wird aber wieder nur eine Instruktion abgearbeitet, danach erfolgt wieder der Trace-Trap. Um in den Normalbetrieb zurückzukehren, muß man das T-Bit in der Kopie des Statusregisters löschen.

Das nebenstehende Assembler-Programm, als 'Single-Step-Monitor' bezeichnet, bietet dem Programmierer die wichtigsten Möglichkeiten zur Überwachung des Programmablaufs im Einzelschrittbetrieb. Mit dem Unterprogramm TRACE werden der Trace-Vektor und der Trap-15-Vektor geladen. Ein Aufruf von Trap 15 schaltet dann in den Single-Step-Modus. Nach Ausführung des darauffolgenden Befehls erscheinen dann in der obersten Bildschirmzeile die Werte und Inhalte des Programmzählers (PC), des User-Stackpointers (USP), des System-Stackpointers (SSP) und des Statusregisters. Der Programmzähler zeigt dabei auf den darauffolgenden Befehl. Die Werte der Stackpointer entsprechen dem Stand vor Aufruf der Trap-Routine.

Der Single-Step-Monitor erlaubt folgende Kommandos:

D – Anzeige der Datenregister

A – Anzeige der Adreßregister

S – Rückkehr in die Programm- und Statuszeile

X – T-Bit löschen

Mit dem Kommando 'X' schaltet man also zurück auf den Normalbetrieb, bei den übrigen drei Kommandos wird jeweils die nächste Instruktion ausgeführt. Um eventuelle Bildschirmausgaben des im Test be-

```
TRACE.LST   *** ASM68000   (c) Herbert Hanewinkel ***

                           ; Single step monitor for ATARI 520ST
                           ;  (c) Herbert Hanewinkel,  Jan. 1986
                      ;--------------------------------------------
00000  487A0026       TRACE:  PEA     STEP    ; load TRACE vector
00004  3F3C0009               MOVE.W  #9,-(A7)
00008  3F3C0005               MOVE.W  #5,-(A7)
0000C  4E4D                   TRAP    #13
0000E  508F                   ADDQ.L  #8,A7
00010  487A0010               PEA     TBIT    ; load TRAP #15 vector
00014  3F3C002F               MOVE.W  #47,-(A7)
00018  3F3C0005               MOVE.W  #5,-(A7)
0001C  4E4D                   TRAP    #13
0001E  508F                   ADDQ.L  #8,A7
00020  4E75                   RTS
                              ;====================
00022  08D70007               TBIT:   BSET    #7,(A7)
00026  4E73                   RTE
                              ;====================
00028  48E7FFFE               STEP:   MOVEM.L #$FFFE,-(A7)
0002C  41FA00AE                       LEA     SAVC,A0
00030  61000100                       BSR     OUT
                              ;---write pc---
00034  41FA00AE               SYS:    LEA     PROG,A0
00038  610000F8                       BSR     OUT
0003C  2A6F003E                       MOVE.L  62(A7),A5
00040  610000E6                       BSR     OUTN
                              ;---write usp---
00044  41FA00A8                       LEA     USTACK,A0
00048  610000E8                       BSR     OUT
0004C  4E6D                           MOVE.L  USP,A5
0004E  610000D8                       BSR     OUTN
                              ;---write ssp---
00052  41FA00A4                       LEA     SSTACK,A0
00056  610000DA                       BSR     OUT
0005A  4BEF0042                       LEA     66(A7),A5
0005E  610000C8                       BSR     OUTN
                              ;---write sr---
00062  41FA009E                       LEA     STAT,A0
00066  610000CA                       BSR     OUT
0006A  3E2F003C                       MOVE.W  60(A7),D7
0006E  610000CE                       BSR     WRIN4
                              ;---command---
00072  41FA0080               INP:    LEA     CLL,A0
00076  610000BA                       BSR     OUT
0007A  3F3C0002                       MOVE.W  #2,-(A7)
0007E  3F3C0002                       MOVE.W  #2,-(A7)
00082  4E4D                           TRAP    #13
00084  588F                           ADDQ.L  #4,A7
00086  0C400053                       CMP.W   #83,D0        ; S ?
0008A  67A8                           BEQ.B   SYS
0008C  0C400044                       CMP.W   #68,D0        ; D ?
00090  6720                           BEQ.B   DRG
00092  0C400041                       CMP.W   #65,D0        ; A ?
00096  6728                           BEQ.B   ARG
00098  0C400058                       CMP.W   #88,D0        ; X ?
0009C  6606                           BNE.B   CON
0009E  08AF0007003C                   BCLR    #7,60(A7)
                              ;-------
000A4  41FA003A               CON:    LEA     RESC,A0
000A8  61000088                       BSR     OUT
000AC  4CDF7FFF                       MOVEM.L (A7)+,#$7FFF
000B0  4E73                           RTE
                              ;---write data-register---
000B2  41FA0058               DRG:    LEA     DATA,A0
000B6  6100007A                       BSR     OUT
000BA  7807                           MOVEQ   #7,D4
000BC  4BD7                           LEA     (A7),A5
000BE  600E                           BRA.B   L11
                              ;---write address-register---
000C0  41FA0056               ARG:    LEA     ADDR,A0
000C4  6100006C                       BSR     OUT
000C8  7806                           MOVEQ   #6,D4
000CA  4BEF0020                       LEA     32(A7),A5
000CE  2E1D                   L11:    MOVE.L  (A5)+,D7
000D0  61000072                       BSR     WRIN8
000D4  51CCFFF8                       DBFA    D4,L11
000D8  6000FF98                       BRA     INP
                              ;--------------------
000DC  1B6A                   SAVC:   DC.B    27,106
000DE  0000                           DC.B    0
000E0  1B6B                   RESC:   DC.B    27,107
000E2  0000                           DC.B    0
000E4  1B48                   PROG:   DC.B    27,72
000E6  50432030D2000                  ASCII   /PC = /
000EC  0000                           DC.B    0
000EE  2020555350203D20       USTACK: ASCII   /  USP = /
000F6  0000                           DC.B    0
000F8  2020535350203D20       SSTACK: ASCII   /  SSP = /
00100  0000                           DC.B    0
00102  20205352203D2000       STAT:   ASCII   /  SR = /
0010A  0000                           DC.B    0
0010C  1B48                   DATA:   DC.B    27,72
0010E  44302D4437203D20               ASCII   /D0-D7 = /
00116  0000                           DC.B    0
00118  1B48                   ADDR:   DC.B    27,72
0011A  41302D4136203D20               ASCII   /A0-A6 = /
00122  0000                           DC.B    0
00124  1B4B                   CLL:    DC.B    27,75
00126  0000                           DC.B    0
                              ;====================
00128  2E0D                   OUTN:   MOVE.L  A5,D7
0012A  6118                           BSR.B   WRIN8
```

```
0012C  3E15                          MOVE.W  (A5),D7
0012E  610E                          BSR.B   WRIN4
00130  4E75                          RTS
                              ;====================
00132  2F08                  OUT:    MOVE.L  A0,-(A7)
00134  3F3C0009                      MOVE.W  #9,-(A7)
00138  4E41                          TRAP    #1
0013A  5C8F                          ADDQ.L  #6,A7
0013C  4E75                          RTS
                              ;====================
0013E  4847                  WRIN4:  SWAP    D7
00140  7C03                          MOVEQ   #3,D6
00142  6002                          BRA.B   L2
00144  7C07                  WRIN8:  MOVEQ   #7,D6
00146  4DFA0034              L2:     LEA     HEX,A6
0014A  E99F                  L1:     ROL.L   #4,D7
0014C  1A07                          MOVE.B  D7,D5
0014E  CABC0000000F                  AND.L   #$F,D5
00154  1A365000                      MOVE.B  0(A6)(D5.W),D5
00158  3F05                          MOVE.W  D5,-(A7)
0015A  3F3C0002                      MOVE.W  #2,-(A7)
0015E  3F3C0003                      MOVE.W  #3,-(A7)
00162  4E4D                          TRAP    #13
00164  5C8F                          ADDQ.L  #6,A7
00166  51CEFFE2                      DBFA    D6,L1
0016A  3F3C0020                      MOVE.W  #32,-(A7)
0016E  3F3C0002                      MOVE.W  #2,-(A7)
00172  3F3C0003                      MOVE.W  #3,-(A7)
00176  4E4D                          TRAP    #13
0017B  5C8F                          ADDQ.L  #6,A7
0017A  4E75                          RTS
0017C  3031323334353637383 9 HEX:    ASCII   /0123456789ABCDEF/
       414243444546
0018C  0000                          END
  Label List :
TRACE                   00
TBIT                    022
STEP                    028
SYS                     034
INP                     072
CON                     0A4
DRG                     0B2
ARG                     0C0
L11                     0CE
SAVC                    0DC
RESC                    0E0
PROG                    0E4
USTACK                  0EE
SSTACK                  0FB
STAT                    0102
DATA                    010C
ADDR                    0118
CLL                     0124
OUTN                    0128
OUT                     0132
WRIN4                   013E
WRINB                   0144
L2                      0146
L1                      014A
HEX                     017C
 396  Bytes saved on file TRACE.OBJ
```

Single-Step-Monitor für den Atari ST

```
1     ' ---------------------------- '
2     ' Execute an assembler program '
3     ' Herbert Hanewinkel, Dec 1985 '
4     ' ---------------------------- '
5     FULLW 2
6     CLEARW 2
10    DIM TRACE%(256)
20    DIM V%(10000) : REM limits maximum program size
30    VA=VARPTR(V%(0)) : A = VA
40    INPUT"Execute program ? ",L$
50    IF LEN(L$) > 6 OR INSTR(L$,".") > 0 THEN 170
60    L$=L$ + ".OBJ"
70    INPUT " Single step mode ? ",Q$
80    IF LEN(Q$) <= 0 OR LEFT$(Q$,1) = "N" THEN 140
90    TA = VARPTR(TRACE%(0))
100   BLOAD "TRACE.OBJ",TA
110   CALL TA
120   A = VARPTR(V%(1))
130   V%(0) = &H4E4F
140   REM === load and execute program ===
150   BLOAD L$,A
160   CALL VA
170   END
180   PRINT " *** Error in filename ***
190   GOTO 40
200   END
```

Dieses BASIC-Programm dient zum Ausprobieren von Assembler-Unterprogrammen.

findlichen Programms nicht unnötig zu stören, wird die Cursor-Position vor jeder Ausgabe abgespeichert und danach wiederhergestellt. Außerdem benutzt der Single-Step-Monitor nur die oberste Programmzeile zur Ausgabe. Der Programmierer kann mit dem Aufruf von Trap 15 an jeder beliebigen Stelle des zu prüfenden Programms in den Einzelschrittbetrieb wechseln. Dieser Aufruf kann beliebig oft im Programm eingesetzt werden.

Ein kleines BASIC-Programm, das zum Ausprobieren von Assembler-Unterprogrammen geschrieben wurde, soll den Einsatz des Single-Step-Monitors demonstrieren. Es lädt das auszuführende Assembler-Programm in den Vektor V%. Auf Wunsch wird der Single-Step-Monitor in den Vektor TRACE% geladen und als erste Instruktion auf dem Vektor V% ein TRAP#15 (&H4E4F) angelegt. Damit wird das nachfolgend geladene Assembler-Programm dann schrittweise ausgeführt. ct

aktuell

'PC-Schreck' wird zum 'PC'

Emulator und zwei neue Modelle von Atari

Atari präsentiert auf der CeBIT in Hannover eine Reihe von Verbesserungen und Ergänzungen seiner erfolgreichen ST-Familie, darunter zwei neue Computer-Modelle und einen Hardware-Zusatz, der den ST in die Lage versetzen soll, den IBM PC zu emulieren.

Die beiden neuen Rechner kommen im 'Apple-IIc-Look': Netzteil und 3,5"- Laufwerk sind in die Konsole eingebaut, die Diskette wird an der rechten Seite eingeschoben. Der Atari 1040 ST entspricht technisch einem 520 ST+ mit 720-KByte-Laufwerk. Der 520 ST FM ist aus einem 260 ST mit 360-KByte-Floppy zusammengesetzt.

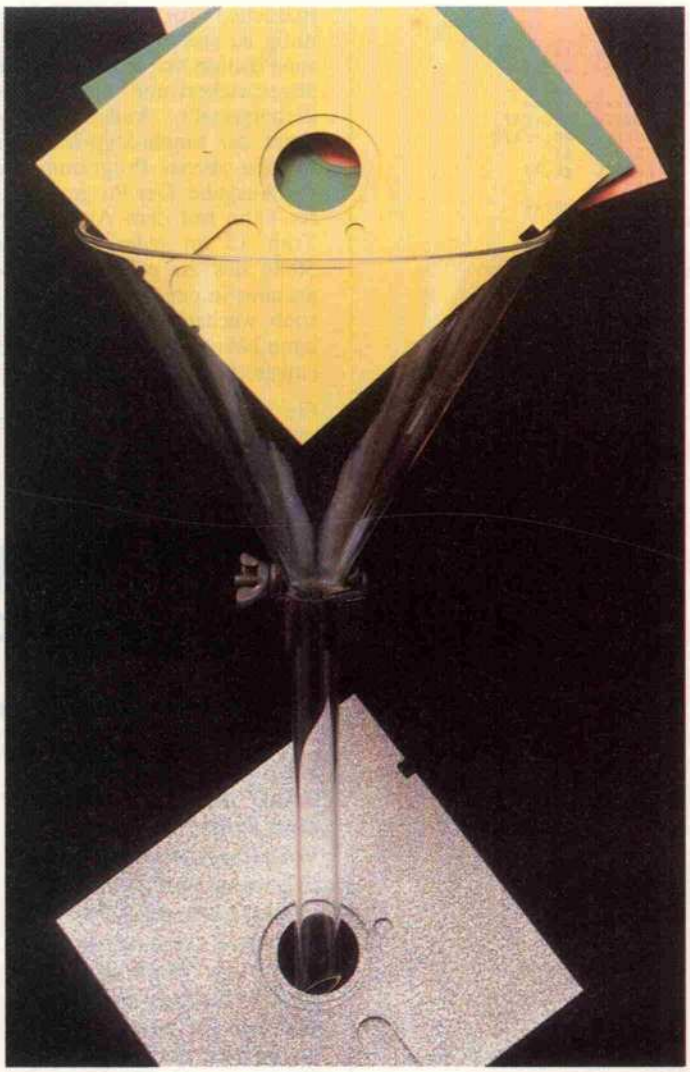
Der PC-Emulator, der für weniger als 1000 Mark in den Handel kommen soll, besteht aus einem Subcomputer in einem getrennten Gehäuse mit eigener Stromversorgung und 512 KByte RAM. Als Prozessor dient ein 8088 mit 8-MHz-Takt, optional unterstützt durch einen Arithmetik-Prozessor 8087. Eine Emulation des Betriebssystems PCDOS wird mitgeliefert; das BIOS befindet sich im ROM auf der Platine. Im Gehäuse ist Platz für den Einbau eines 5,25"-Laufwerks vorgesehen; Software zum Transfer von Dateien zwischen 'PC' und 'ST' soll ebenfalls zum Lieferumfang gehören. Die Verbindung zur Rechnerkonsole erfolgt über den DMA-Anschluß. Im 'ST-Modus' steht dem Rechner der zusätzliche Arbeitsspeicher noch als RAM-Floppy zur Verfügung.

Als weiteres neues Produkt will Atari eine Grafik-Erweiterung zur nachträglichen Umrüstung der ST-Modelle präsentieren. Deren Farbgrafik-Darstellung wird damit der des Commodore Amiga ebenbürtig: 640 × 400 Pixel in 16 Farben aus einer Palette von 4096. Die Umrüstung soll bei jedem Atari-Fachhändler durch Einbau einer kleinen Zusatzkarte vorgenommen werden können.

Rechtzeitig zur Messe soll auch die lang erwartete ROM-Version des Betriebssystems in deutscher Sprache fertig sein, die beim Übersetzen zunächst etwas zu lang geraten war. ST-Besitzer können die ROMs zum Preis von 99 Mark bei ihrem Atari-Händler einsetzen lassen. Auch die Auslieferung des Winchester-Laufwerks soll ab Mitte März beginnen. Der häufig kritisierte BASIC-Interpreter wird nach Angaben von Atari nicht weiter korrigiert. Statt dessen soll jetzt ein neuer Interpreter ausgeliefert werden, der einen Umfang von 80 KByte besitzt und von Metacomco entwickelt wurde. ct

Hans-Georg Joepgen

Der in der Programmierkunst Fortgeschrittene fängt nicht jedesmal bei Adam und Eva an, sondern verwendet zur Lösung neuer Aufgaben auch vorfabrizierte und ausgetestete Programmbausteine, sogenannte Module. Leider kommen die gebräuchlichen Disketten-Betriebssysteme dem Umgang mit Modulen nicht entgegen, weil sie beim Abspeichern kleiner Bausteine höchst unwirtschaftlich mit der Disketten-Kapazität umgehen. Die hier vorgestellten Dienstprogramme benutzen ein einfaches Verfahren zur Datenkompression und fassen mehrere Module in einer Bibliothek zusammen. So läßt sich beispielsweise auf einer Diskette, auf der nur 128 Module Platz hätten, der Text von über 1000 Modulen unterbringen.



Tausend Programme pro Diskette

Modulbibliotheken unter CP/M-80 und MS-DOS

Das Speichervermögen eines Datenträgers wird von drei Größen begrenzt. Diese sind seine in Bytes gemessene Kapazität, die maximale Anzahl der möglichen Directory-Einträge und schließlich die Anzahl der Blöcke, also der Gruppen aus benachbarten Sektoren, die das Betriebssystem als Einheit behandelt. Legt man etwa eine kleine Datei aus 100 Bytes auf einer Diskette mit einer Blockgröße von 4 KByte ab, so wird dadurch ebensoviel Platz verbraucht, als habe man 4096 Bytes abgespeichert.

Oder ein anderes Beispiel: Wer 100 kleine Programm-Module archivieren möchte, die auf einer Apple-CP/M-Diskette eigentlich bequem Platz fänden, braucht dazu dennoch drei Disketten, wenn jede nur 48 Directory-Einträge zuläßt. Ist erst einmal der für das Inhaltsverzeichnis vorgesehene Platz belegt, erscheint die Diskette als voll, auch wenn auf ihr noch viele zehntausend Bytes an ungenutzten Sektoren übrigbleiben.

Schließlich wird weiterer Speicherplatz dadurch verschwendet, daß zur Abspeicherung eines ASCII-Zeichens ein komplettes Byte belegt wird, obwohl zur Kennzeichnung eines ASCII-Zeichens sieben Bit vollständig ausreichen.

Doppelte Maßnahme

Bei der Archivierung von kleinen Modulen wirken sich diese Eigenarten der Betriebssysteme nicht gerade dämpfend auf die Diskettenkosten und den Platzbedarf des Archivs aus. Zwei Programme in Turbo-Pascal ermöglichen eine wirtschaftliche Kapazitätsnutzung. Sie sind in Turbo-Kernel, also der den Versionen eins bis drei gemeinsamen Sprache geschrieben und laufen nach geringfügigen Änderungen (procedure betriebsart_ermittlung) sowohl unter CP/M-80 und CP/M-86 als auch unter MS-DOS.

Die Programme sind in der Lage, Text-Files zu komprimieren, in einer Bibliotheksdatei zusammenzufassen und natürlich die ursprünglichen Text-Files wiederherzustellen. Die Kompression geschieht, indem jedes achte Zeichen einer Zeile auf die achten Bits der vorangehenden sieben Bytes aufgeteilt wird. Mit Hilfe dieser 'Sieben-Achtel-Methode' läßt sich bis zu 12,5 % mehr Text auf dem Datenträger unterbringen. Die Zusammenfassung kleiner Module in einer Bibliotheksdatei spart weiteren Diskettenplatz und Directory-Einträge. Das Programm BIBMAKER erzeugt die komprimierten Bibliotheksdateien, und PACKAUS3 gewinnt die ursprünglichen Text-Files zurück.

BIB-Dateien

BIBMAKER liest die einzelnen Module über die Eingabedateien 'fi', legt über 'fo' die komprimierte Bibliotheksdatei, die BIB-Datei, an und bedient sich gegebenenfalls noch einer Steuerdatei 'fc'. Gibt man nämlich beim Aufruf von BIBMAKER noch einen Parameter an, erwartet das Programm die Steuerdatei PSEND.CMF, die eine Tabelle der zu komprimierenden Module enthalten muß. Der Aufruf von 'BIBMAKER TEST' führt beispielsweise dazu, daß die in PSEND.CMF aufgezählten Module in der Bibliotheksdatei TEST.BIB zusammengefaßt werden. Das Programm versieht die Bibliotheksdateien also automatisch mit der Typbezeichnung 'BIB'.

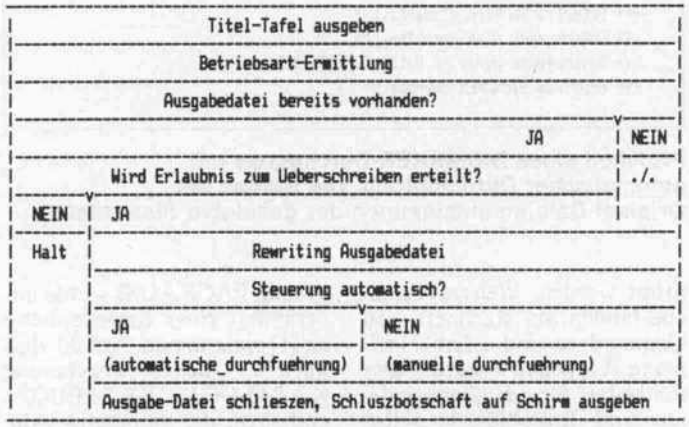
Die Steuerdatei PSEND.CMF kann zum Beispiel mit dem Turbo-Editor erzeugt werden. Die einzelnen Modul-Namen müssen jeweils durch einen Zeilenwechsel (CR/LF) getrennt sein, wie das Hex-Listing zeigt.

Mit Aufrufparameter schaltet BIBMAKER also auf die Betriebsart 'automatisch'. Sie empfiehlt sich besonders dann, wenn eine BIB-Datei häufig ergänzt werden muß. In einem solchen Fall genügt es, nur die Steuerdatei PSEND.CMF zu

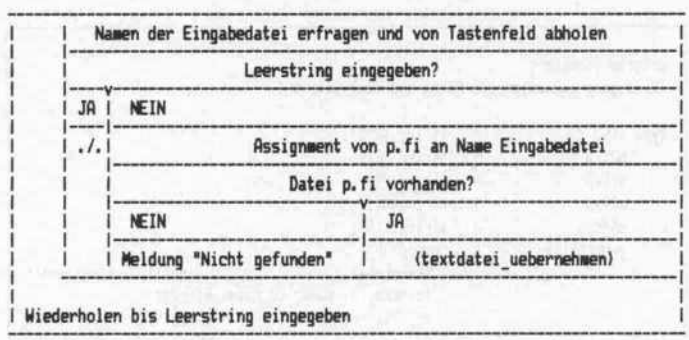
ergänzen, alles weitere geschieht dann automatisch. Ohne Aufrufparameter erfragt das Programm im Dialog die Module, die komprimiert und zusammengefaßt werden sollen.

Auspacken

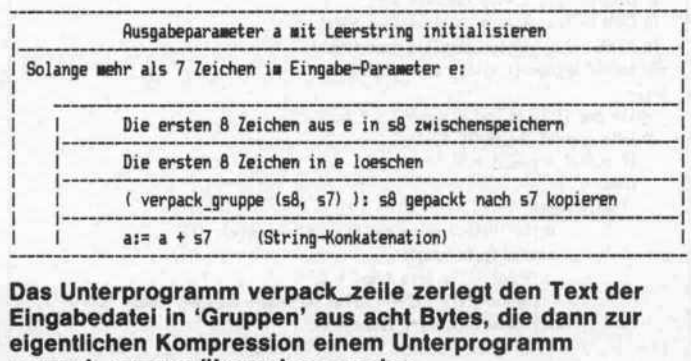
Das Programm PACKAUS3 besorgt die Rückübersetzung der ASCII-Zeichen aus dem 'Sieben-Achtel-Format' in Bytes und erzeugt Kopien der ursprünglichen Textdateien. Deren Namen entnimmt PACKAUS3 speziellen Steuerzeilen innerhalb der BIB-Dateien; diese Steuerzeilen beginnen mit drei Dollarzeichen. Deswegen sind Zeilen mit 'Drei-Dollar-Anfängen' in den Original-Textdateien verboten.



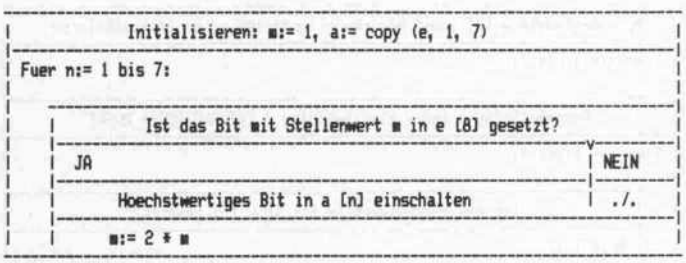
Hauptprogramm zu BIBMAKER: Nach Ausgabe der Titeltafel wird durch Aufruf eines Unterprogramms BETRIEBSART_ERMITTLUNG die 'General-Variable' V initialisiert. In ihr führt BIBMAKER unter anderem Buch über Informationen wie den Namen der anzufertigenden Bibliotheksdatei. Je nach vorgewählter Betriebsart übernimmt sodann eine der beiden Prozeduren automatische_durchführung oder manuelle_durchführung die 'Knochenarbeit' beim BIB-Datei-Bau.



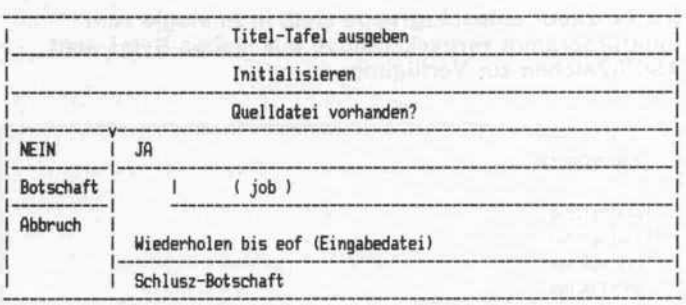
Die Prozedur manuelle_durchführung liest Dateien, deren Namen über das Tastenfeld eingegeben werden, und baut sie in die zum Zeitpunkt des Aufrufs bereits existierende BIB-Datei ein.



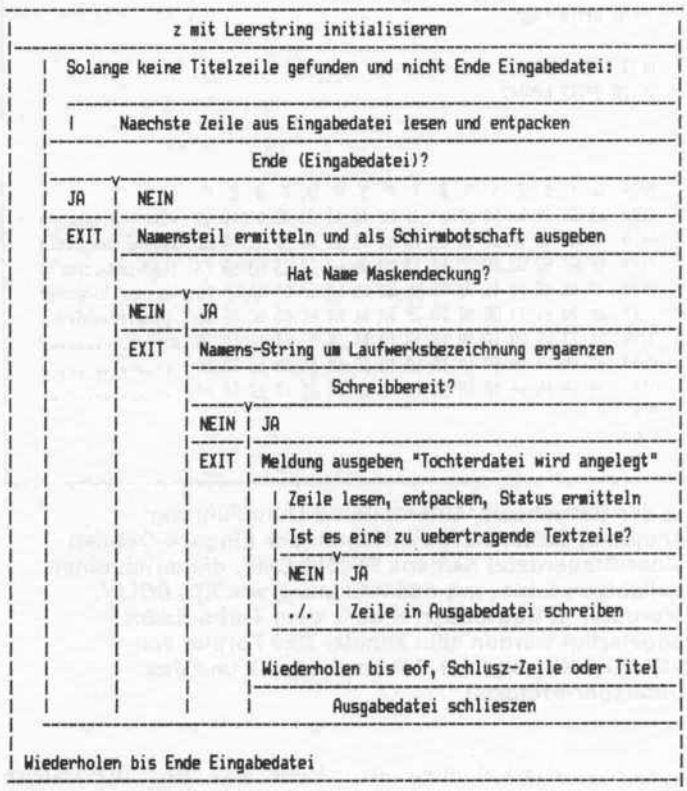
Das Unterprogramm verpack_zeile zerlegt den Text der Eingabedatei in 'Gruppen' aus acht Bytes, die dann zur eigentlichen Kompression einem Unterprogramm verpack_gruppe übergeben werden.



Endstation: In der Prozedur verpack_gruppe erfolgt die Umschlüsselung von acht ASCII-Zeichen in sieben Bytes. Die Variable m dient dazu, die sieben niederwertigen Bits des achten Gruppenbytes abzutasten, die dann in die höchstwertigen Bits der anderen sieben Gruppenmitglieder eingeschlüsselt werden.



Mit dem Programm PACKAUS3 werden den komprimierten BIB-Dateien die einzelnen Textdateien entnommen und in der ursprünglichen Form auf Datenträger geschrieben.



Job ist die Hauptroutine von PACKAUS3. Dieses Unterprogramm analysiert BIB-Dateien, prüft, ob die in die BIB-Datei verschlüsselten Namen der ursprünglichen Original-Files einer Auswahl-Maske entsprechen, und stellt die Original-Dateien wieder her.

```
| m, Laengenbyte a [0] sowie String-Schluszglied a [8] initialisieren |
|---------------------------------------------------------------------|
| Fuer n:= 1 bis 7:                                                   |
|                                                                     |
|   ------------------------------------------------------------------|
|   |    Ausgabestring a [n]:= Eingabestring e [n] AND ASCII_MASKE    |
|---------------------------------------------------------------------|
| Fuer n:= 1 bis 7:                                                   |
|                                                                     |
|   ------------------------------------------------------------------|
|   |          Ist das hoechstwertige Bit in e [n] gesetzt?           |
|   |------v-----------------------------------------------------------|
|   | NEIN | JA                                                        |
|   |------|-----------------------------------------------------------|
|   | ./. | a [8]:= a [8] OR M    {Bit mit Stellenwert m in a [8] setzen} |
|   |-------------------------------------------------------------------|
|   |                        m:= 2 * m                                  |
|---------------------------------------------------------------------|
|       Ist das letzte Zeichen im Ausgabestring a nun ASCII 'NUL'?    |
|-----v---------------------------------------------------------------|
| NEIN | JA                                                           |
|------|--------------------------------------------------------------|
| ./. | Letztes Zeichen im Ausgabestring a entfernen                   |
-----------------------------------------------------------------------
```

Die Prozedur entpack_gruppe stellt in Analogie zum Unterprogramm verpack_gruppe aus sieben Bytes acht ASCII-Zeichen zur Verfügung.

```
P) BIBMAKER TESTBIB

====================
BIBMAKER (Joe/8-6-85)
====================

Erzeugung einer gepackten BIB-Datei aus Text-Dateien.

+++ Uebertragen wird katalog1.pas
+++ Uebertragen wird katalog2.pas
+++ Uebertragen wird katalog3.pas
+++ Uebertragen wird checksum.pas
+++ Uebertragen wird bibmaker.pas
+++ Uebertragen wird przahlen.pas
+++ Geschrieben wurde in Datei TESTBIB.BIB.
+++ Programm BIBMAKER beendet.-
```

Protokoll eines BIBMAKER-Durchgangs mit automatischer Durchführung. Die Namen der Original-Dateien entstammen der gezeigten Steuerdatei.

```
TYPE PSEND.CMF

katalog1.pas
katalog2.pas
katalog3.pas
checksum.pas
bibmaker.pas
przahlen.pas

P) T80 PSEND.CMF

TRACE80 V2.06-00106
COPYRIGHT (C) 1981,82 BY L. LAUTERBACH
35584 BYTES FREE

NEXT FREE 0180
00128 BYTES LOADED

* D

ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0100  6B 61 74 61 6C 6F 67 31 2E 70 61 73 0D 0A 6B 61  katalog1.pas..ka
0110  74 61 6C 6F 67 32 2E 70 61 73 0D 0A 6B 61 74 61  talog2.pas..kata
0120  6C 6F 67 33 2E 70 61 73 0D 0A 63 68 65 63 6B 73  log3.pas..checks
0130  75 6D 2E 70 61 73 0D 0A 62 69 62 6D 61 6B 65 72  um.pas..bibmaker
0140  2E 70 61 73 0D 0A 70 72 7A 61 68 6C 65 6E 2E 70  .pas..przahlen.p
0150  61 73 0D 0A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A  as..............
0160  1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A  ................
0170  1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A 1A  ................
0180 -
* BYE
```

In der Betriebsart 'Automatische Durchführung' entnimmt BIBMAKER die Namen der Eingabe-Dateien einer Steuerdatei namens PSEND.CMF, die durch einen beliebigen Editor mit ASCII-Eignung wie ED, EDLIN, WordStar in Betriebsart N oder dem Turbo-Editor angefertigt worden sein konnte. Das Format von PSEND.CMF zeigt der Schirmausdruck und das Debugger-Protokoll.

Es werden je nach Wunsch entweder alle in einer BIB-Datei verschlüsselten Files in der ursprünglichen Gestalt wiederhergestellt oder nur jene, deren Namen von einer Auswahlmaske erfaßt werden. Dadurch kann man den BIB-Dateien auch selektiv einzelne Module entnehmen.

Das Listing von BIBMAKER zeigt, auf welchen Rechnern und unter welchen Betriebssystemen beide Programme erprobt wurden. Weitere Versuche fanden auf Rechnern von Commodore und ELab Hoffmann-Rappenau statt. Der Zeitbedarf für das Komprimieren und Entschlüsseln selbst großer Dateien ist gering. Selbst Datenmengen im Umfang mehrerer hundert Kilobyte sind in wenigen Minuten übersetzt.

Bei der Entwicklung und Erprobung beider Programme waren die Fachleute Hubert Palme, Ulrich Schmidt und Matthias Joepgen behilflich. Das Programm PACKAUS3 wurde unverändert einer Begleitdiskette zu [1] entnommen. Auf ihr sind auch die Programme KATALOG1. . .KATALOG3 enthalten, die automatisch die Steuerdatei PSEND.CMF durch Eingabe einer Auswahlmaske erzeugen können.

Literatur

[1] Joepgen, Hans-Georg: Turbo-Pascal; das Kompendium für die Programmier-Praxis. Carl Hanser Verlag, München, Wien; 1985.

```
program bibmaker;
{Erzeugung gepackter BIB-Datei aus Text-Dateien}

type styp           =  string [255];
     s2typ          =  string [2];
     s7typ          =  string[7];
     s8typ          =  string[8];
     s14typ         =  string [14];
     generaltyp     =  record
                         steuerung:                (manuell, automatisch);
                         fc_name, fi_name, fo_name: s14typ;
                         fc, fi, fo:                text;
                         c:                         char
                       end;
var  v: generaltyp;

procedure betriebsart_ermittlung (var p: generaltyp);
{Definition von aufruf_argument ist betriebssystemabhaengig. Setzen Sie bitte
 in CP/M-80 '... s14typ absolute $80;'
 in CP/M-86 '... s14typ absolute dseg:$80;'
 in MSDOS   '... s14typ absolute cseg:$80;'}
var aufruf_argument: s14typ absolute $80;
begin
  while pos (' ', aufruf_argument) = 1 do
  delete (aufruf_argument, 1, 1);
    if aufruf_argument = ''
    then
      begin repeat
              write ('+++ Vorname der Bibliotheks-Datei: ');
              readln (p.fo_name);
                if pos ('.', p.fo_name) ) 0
                then writeln (^G'--- Keine Nachnamen angeben. Immer "BIB".-');
            until (pos ('.', p.fo_name) = 0)
```

```
                and (p.fo_name ) '');
        p.fo_name:= p.fo_name + '.BIB';
        p.steuerung:= manuell
    end
  else begin
        p.fo_name:=    aufruf_argument;
          if pos ('.', p.fo_name) ) 0
          then begin
                writeln (^G'--- Keine Nachnamen angeben. Immer "BIB".');
                writeln ('    Programm wird abgebrochen.-');
                halt
               end;
        p.fo_name:= p.fo_name + '.BIB';
        p.fc_name:= 'psend.cmf';
        p.steuerung:=  automatisch
       end;
end;

function vorhanden (var datei: text): boolean;
begin
  {$I-} reset (datei) {$I+};
  vorhanden:= ioresult = 0
end;

procedure verpack_gruppe (e: s8typ; var a: s7typ);
const msb: byte = $80;
var m, n: byte;
begin
  m:= 1; a:= copy (e, 1, 7);
  for n:= 1 to 7 do
  begin
    if (byte (e [8]) and m) ) 0
    then a [n]:= char (byte (a [n]) or msb);
    m:= m * 2
  end;
end;

procedure verpack_zeile (var e, a: styp);
var s7: s7typ; s8: s8typ;
begin
  a:= '';
  while length (e) ) 7 do begin
                          s8:= copy (e, 1, 8);
                          delete (e, 1, 8);
                          verpack_gruppe (s8, s7);
                          a:= a + s7
                         end;
  a:= a + e
end;

procedure programm_abbruch (botschaft: styp; var fo: text);
begin
  writeln (^G'--- ', botschaft, '.');
  writeln (^G'    Programm wird abgebrochen, Ausgabedatei geloescht.-');
  close (fo);
  erase (fo);
  halt
end;

procedure textdatei_uebernehmen (var p: generaltyp);
var s, ss: styp;  n: integer;
begin
  assign (p.fi, p.fi_name);
    if not vorhanden (p.fi)
    then programm_abbruch ('Datei ' + p.fi_name + ' nicht gefunden', p.fo);
    if eof (p.fi)
    then programm_abbruch ('Datei ' + p.fi_name + ' ist leer', p.fo);
  s:= p.fi_name; n:= pos (':', s);
    if n) 0 then delete (s, 1, n);
  s:= '$$$ ' + s;
  verpack_zeile (s, ss);
  writeln (p.fo, ss);
  repeat
    readln (p.fi, s);
    verpack_zeile (s, ss);
    writeln (p.fo, ss)
  until eof (p.fi);
  close (p.fi);
  s:= '             {----------------------------------------------}';
  verpack_zeile (s, ss);
  writeln (p.fo, ss)
end;

procedure manuelle_durchfuehrung (var p: generaltyp);
begin
  repeat
    write ('+++ Welche Datei uebernehmen ((CR): Abbruch): ');
    readln (p.fi_name);
      if p.fi_name ) ''
      then begin
             assign (p.fi, p.fi_name);
               if not vorhanden (v.fi)
               then writeln (^G'--- Datei ' + v.fi_name + ' nicht gefunden.-')
               else textdatei_uebernehmen (p);
           end;
  until p.fi_name = ''
end;

procedure automatische_durchfuehrung (var p: generaltyp);
begin
  assign (p.fc, p.fc_name);
    if not vorhanden (p.fc)
    then programm_abbruch
         ('Steuerdatei ' + p.fc_name + ' nicht gefunden', p.fo);
   repeat
     readln (p.fc, p.fi_name);
     writeln ('+++ Uebertragen wird ', p.fi_name);
     textdatei_uebernehmen (p)
   until eof (p.fc)
end;

begin {picmaker}
  clrscr;
  writeln ('====================');
  writeln ('BIBMAKER (Joe/8-6-85)');
  writeln ('====================^J);
  writeln ('Erzeugung einer gepackten BIB-Datei aus Text-Dateien.'^M^J);
  betriebsart_ermittlung (v);
  assign (v.fo, v.fo_name);
    if vorhanden (v.fo)
    then repeat
           write ('--- Datei ', v.fo_name, ' existiert. ');
           write  ('Ueberschreiben (J/N)? ');
           read (kbd, v.c); v.c:= upcase (v.c);  writeln (v.c);
            if not (v.c in ['J', 'N']) then write (^G);
         until v.c in ['J', 'N'];
         if v.c = 'N' then halt;
  rewrite (v.fo);
    if v.steuerung = automatisch
    then automatische_durchfuehrung (v)
    else manuelle_durchfuehrung (v);
  close (v.fo);
  writeln ('+++ Geschrieben wurde in Datei ',v.fo_name, '.');
  writeln ('+++ Programm BIBMAKER beendet.-')
end.
```

Das Programm BIBMAKER spart Diskettenplatz: Es wendet ein einfaches Kompressionsverfahren an und faßt mehrere kleine Text-Module zu einer Bibliotheksdatei zusammen.

```
program packaus_3 {Version 3.3.85, Fassung fuer Begleitdiskette 3};
{
Revisions-Stand:
 3.3.85: Depacking eingebaut.
24.1.85: Wordstar- und TLIST-Punktkommandos ausgeblendet, Korrektur an
         Programmantwort bei Fehlbedienung Namenseingabe.
18.1.85: Wenn Dateiname kuerzer als voll spezifizierte Suchmaske, erfolgte
         bei manchen Mustern faelschlich Auswahl. MASKENDECKUNG korrigiert.
17.1.85: Freigabetest unter CP/M 2.2, CP/M-Plus, MS-DOS 2.11, CP/M-86,
         PCDOS 2.01, Concurrent CP/M 3.1, Concurrent DOS 3.2 und MP/M II
         mit Turbo-Versionen 2.1 (CP/M-86) und 2.0 (uebrige Betriebssysteme)
```

```
        auf Apple II Plus, Apple II E, ITT 2020, ITT 3030, IBM PC-XT
        und Wave Mate Bullet IV/Joe.
17.1.85: WARNEN als Option bei Existenz Tochterdatei eingebaut.
16.1.85: Default-Problem auf IBM PC/XT mit frueher ROS-Version behoben.
14.1.85: Maskenbetrieb und Laufwerks-Waehlbarkeit eingebaut.
 7.1.85: Um Zusatz-Endekennung wegen Modulen ohne 'end.'-Kenner erweitert.
 3.1.85: Vorversuche abgeschlossen. Prototyp-Version laeuft.
}
type laufwerksnamen_typ = string [2];
     statustyp          = (neutralzeile, titelzeile, schlusszeile);
     zeilentyp          = string [100];
     namenstyp          = string [14];
     betriebsart_typ    = record warnen:        boolean;
                                 ziel_laufwerk:  laufwerksnamen_typ;
                                 quelldateiname,
                                 maske:          namenstyp
                          end;
     s7typ = string[7];
     s8typ = string[8];
     styp  = string[255];

var fi: text;  betriebsart: betriebsart_typ;

procedure entpack_gruppe (e: s7typ; var a: s8typ);
const msb: byte = $80; ascii_maske = $7f;
var m, n: byte;
begin
  m:= 1; a[0]:= char (8); a [8]:= char (0);
  for n:= 1 to 7 do a [n]:= char (byte (e [n]) and ascii_maske);
  for n:= 1 to 7 do
  begin
    if (byte (e [n]) and msb) ) 0
    then a [8]:= char (byte (a [8]) or m);
    m:= m * 2
  end;
  if a [8] = char (0)
  then delete (a, 8, 1);
end;

procedure entpack_zeile (var e, a: styp);
var s7: s7typ; s8: s8typ;
begin
  a:= '';
  while length (e) ) 6 do begin
                            s7:= copy (e, 1, 7);
                            delete (e, 1, 7);
                            entpack_gruppe (s7, s8);
                            a:= a + s8;
                          end;
  a:= a + e
end;

procedure initialisierung (var p: betriebsart_typ);
var s: namenstyp; n: byte; eingabe_richtig: boolean;
begin
  writeln
  (^M^J'-----------------------------------------------------------------');
  repeat
    write   ('+++ Vorname Mutterdatei ((cr): "DOSCALLS"): ..................... ');
    readln (s);
    eingabe_richtig:= pos ('.', s) = 0;
      if not eingabe_richtig
      then writeln (^G'    Nachnamen bitte nicht angeben.');
  until eingabe_richtig;
    if s = '' then p.quelldateiname:= 'doscalls.bib'
              else p.quelldateiname:= s + '.bib';
  repeat
    write ('+++ Ziel-Laufwerk fuer Tochterdateien ((cr): Vorwahl-LW): ..... ');
    readln (s);
      if copy (s, 2, 1) = ':' then delete (s,2,1);
    eingabe_richtig:=
    (s = '') or ((length (s) = 1) and (s [1] in ['A'..'P', 'a'..'p']));
      if not eingabe_richtig then writeln (^G);
  until eingabe_richtig;
    if s () '' then s:= s + ':';
  p.ziel_laufwerk:= s;
  write ('+++ Auswahl-Maske Vornamen der Tochter-Dateien ((cr): "*"): ..... ');
  repeat
    readln (s);
    eingabe_richtig:= (s = '') or ((pos (s, '.') = 0) and (length (s) (= 8));
      if not eingabe_richtig then write (^G);
  until eingabe_richtig;
    for n:= 1 to length (s) do s [n]:= upcase (s [n]);
    while length (s) ( 8 do s:= s + '?';
  p.maske:= s;
  write ('+++ Toechter ohne Warnung ueberschreiben (J/N; (cr) = "J"): ..... ');
  repeat
    readln (s); s [1]:= upcase (s [1]);
    eingabe_richtig:= (s = '') or (s = 'J') or (s = 'N');
      if not eingabe_richtig then write (^G);
  until eingabe_richtig;
  betriebsart.warnen:= (s = 'N');
  writeln ('-----------------------------------------------------------------')
end;

function maskendeckung (n: namenstyp; b: betriebsart_typ): boolean;
var index: byte; zeichendeckung: boolean;
begin
  if b.maske = '????????' then maskendeckung:= true
  else begin
         index:= 0;
           if pos ('.', n) ) 0 then n:= copy (n, 1, pos ('.', n) - 1);
         while length (n) ( 8 do n:= n + ' ';
         repeat
           index:= index + 1;
           zeichendeckung:= (upcase (n [index]) = b.maske [index])
                            or (b.maske [index] = '?')
                            or (b.maske [index] = '*')
         until (not zeichendeckung)              {Diskrepanz gefunden}
               or (index = length (n))           {Alle Zeichen abgetastet}
               or (b.maske [index] = '*')        {Stern deckt Folgezeichen ab}
               or (n [index + 1] = '.');         {Vorname von N fertig}
         maskendeckung:= zeichendeckung
       end {else};
end {maskendeckung};

function zeilenstatus (p: zeilentyp): statustyp;
begin
  if pos ('end.', p) = 1
  then zeilenstatus:= schlusszeile
  else if pos ('$$$', p) = 1
       then zeilenstatus:= titelzeile
       else zeilenstatus:= neutralzeile
end;

function namensteil (p: zeilentyp): namenstyp;
begin delete (p, 1, 4); namensteil:= p end;

function schreibbereit
(var fo:text; nn: namenstyp; warnung_gefordert: boolean): boolean;
{Ordnet FO dem Namen NN zu, holt Ueberschreibgenehmigung, wenn WARNFLAG
gesetzt ist; uebergibt FALSE oder TRUE und eroeffnete Datei FO}
label exit; var c: char;
begin
  assign (fo, nn);
    if warnung_gefordert
    then begin
           {$i-} reset (fo) {$i+};
           if ioresult = 0
           then begin
                  write
                  ('*** Datei ', nn, ' existiert. Ueberschreiben (J/N)? ');
                  repeat
                    read (kbd, c); c:= upcase (c);
                      if not (c in ['J', 'N'] )then write (^G);
                  until c in ['J', 'N'];
                  writeln;
                    if c = 'N'
                    then begin
                           schreibbereit:= false;
                           goto exit
                         end
                end {ioresult = 0}
         end {warnung_gefordert};
  {$i-} rewrite (fo) {$i+};
    if ioresult ) 0
```

```
    then begin
           writeln (^G'))) Tochterdatei ', nn, ': REWRITE-Fehler');
           schreibbereit:= false
         end
    else schreibbereit:= true;
  exit:
end;

function zeile: styp;
{Liest aus Datei fi und uebergibt dekodierte Zeile}
var e, a: styp;
begin readln (fi, e); entpack_zeile (e, a); zeile:= a end;

procedure job;
{Uebernimmt Mutterdatei fi geoeffnet, erzeugt und fuellt Tochterdateien}
label exit;
var z: zeilentyp; s: statustyp; n: namenstyp; fo: text;
begin
  z:= '';
  repeat
    while (zeilenstatus (z) () titelzeile) and not eof (fi) do z:= zeile;
      if eof (fi) then goto exit;
    n:= namensteil (z); writeln ('+++ in Mutter-Datei gefunden: ', n, '.-');
      if not maskendeckung (n, betriebsart) then goto exit;
    n:= betriebsart.ziel_laufwerk + n;
      if not schreibbereit (fo, n, betriebsart.warnen) then goto exit;
    writeln ('--) Tochterdatei wird angelegt.');
      repeat z:= zeile; s:= zeilenstatus (z);
        if (s () titelzeile)
          and (pos ('.', z) () 1)  {Wordstar-Punktkommando}
          and (pos ('(.', z) () 1)  {TLIST.Punktkommando}
          then writeln (fo, z)
      until (s = schlusszeile) or (s = titelzeile) or eof (fi);
    close (fo)
  until eof (fi);
  exit:
end;

begin {packaus 3} writeln (^M^J^M^J^M^J^M^J^M^J^M^J^M^J);
  writeln ('======================');
  writeln ('PACKAUS 3 (Joe/3.3.85)');
  writeln ('======================'^M^J);
  writeln
  ('Dieses Programm ist Teil des Buches  "T u r b o - P a s c a l"');
  writeln
  ('und seiner "Begleitdiskette 3" von Hans-Georg Joepgen, erschienen im');
  writeln ('Carl Hanser Verlag, Muenchen und Wien.');
  writeln;
  writeln
  ('PACKAUS dient zur Erzeugung von Tochterdateien aus DOSCALLS.BIB oder');
  writeln
  ('anderen Mutter-Dateien mit dem Nachnamen ".BIB" ("Baustein-BIBliothek").');
  writeln ('Dabei kann das Ziel-Laufwerk angegeben werden (Ersatzwert ist das');
  writeln ('vorgewaehlte Laufwerk).-');
  writeln;
  writeln ('PACKAUS 3 entpackt Dateien mit und ohne Daten-Konzentration.');
  writeln;
  writeln ('Bei Betriebsstoerungen wie "Datentraeger-Kapazitaet erschoepft"');
  writeln ('oder "Laufwerk nicht betriebsbereit" bricht Turbos "I/O-Waechter"');
  writeln ('im Kompilat das Programm ab. Die Bedeutung der dann ausgegebenen');
  writeln ('Fehler-Kennung ersehen Sie aus der "I/O-Fehler-Tabelle" Ihres');
  writeln ('Turbo-Originalhandbuches.-');
  initialisierung (betriebsart);
  assign (fi, betriebsart.quelldateiname); {$i-} reset (fi) {$i+};
    if ioresult ) 0
    then begin
           writeln
           (^M^J^G'*** ', betriebsart.quelldateiname, ' nicht gefunden.');
           writeln ('    Programm PACKAUS 3 deswegen abgebrochen.-');
           halt
         end;
  repeat job until eof (fi);
  writeln (^M^J'+++ Programm PACKAUS 3 beendet.-')
end.
```

Das Programm PACKAUS3 packt aus: Es entnimmt die Text-Bausteine wieder der Bibliotheksdatei – auch selektiv – und stellt sie in ihrer ursprünglichen Form wieder her. c't

What's the second best thing you can do for your system?

ACS KEYBOARDS

Modell AN111FXT DM 526,00 ohne MWSt. (DM 599,64 incl. MWSt.)
Das Keyboard für deutsche Textverarbeitung, einsteckfertig, passend für PC/XT-Kompatible (AT in Vorbereitung)

kompatible

Zusätzliche Leistungsmerkmale:
— 2fach PROG-Modus für 10 Fixprogrammiertasten
— Abgesetzter 10er Block, große RETURN & ENTER Taste
— 9 LEDs für CAPS-LOCK, NUM-LOCK, PROG usw.
— 7 weitere sinnvolle Funktionstasten, 2 Paßwörter

Modell AN96FI DM 622,00 ohne MWSt. (709,08 incl. MWSt.)
Das hochintelligente Keyboard für den professionellen Entwickler und Programmierer mit 16 Funktionstasten. Anpaßbar an alle 7/8 Bit-Systeme.

intelligent

Sämtliche nur erdenkliche Möglichkeiten wie:
— Variable Stringausgabe mit QUIT-Marken
— 80 Tasten freiprogrammierbar mit Datenerhalt
— 4 Hauptfunktionsebenen umschaltbar mit 4 LEDs
— Paßwort, BELL, Tastenclick und v. m.
— SKIP für 8. Datenbit, Downloading usw.

Modell AN111FU DM 540,00 ohne MWSt. (DM 615,60 incl. MWSt.)
PROFI-Tastatur deutsche Textverarbeitung DIN 2137 mit 32 Funktionstasten

universell

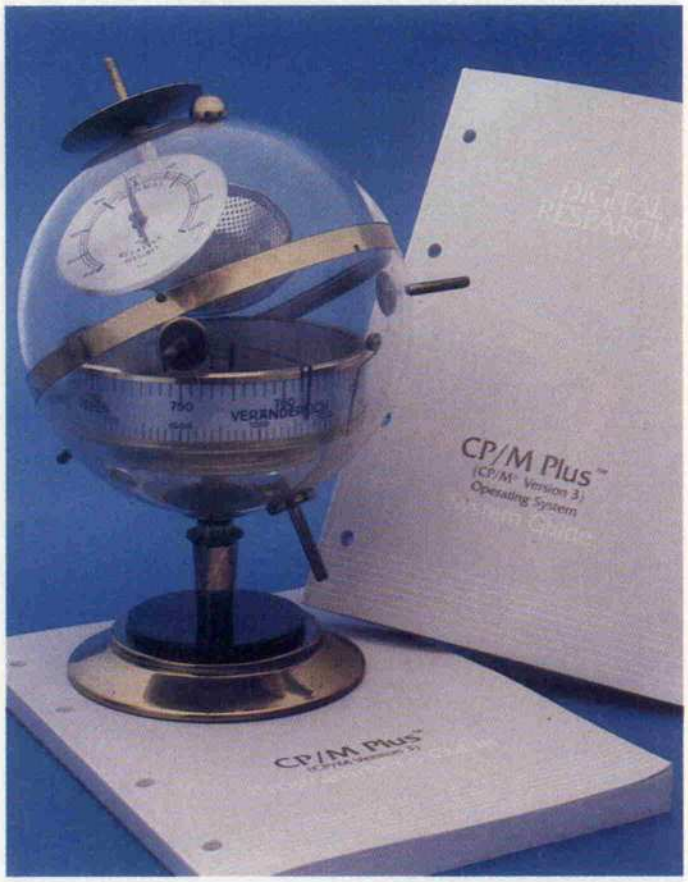
Folgende herausragende Standardleistungsmerkmale:
— Serielle und parallele Schnittstelle
— STRINGS beliebiger Länge für jede Taste
— Master-Paßwort und Security-Paßwort
— 3 Hauptfunktionsebenen mit 3 LEDs
— 9 LEDs für die wichtigsten Betriebszustände
— 2-fach PROG-MODUS für Fixprogrammiertasten

the best for your system!

ACS GmbH

steuerungssysteme
computerperipherie
datentechnik

D-4930 Detmold
Alter Mühlenweg 5
Telefon 05231-41 76/41 77
Telex 935660 acs d



Veränderlich

Systemerweiterungen unter CP/M-Plus

Hans-Peter Sauer

Eine der vielen neuen Eigenschaften von CP/M-Plus ist die Möglichkeit, den Betriebssystemkern modular zu erweitern. Hierzu dienen die 'festen Systemerweiterungen', original 'Resident System Extensions' (RSX) genannt. Eine RSX erlaubt dem Programmierer, eine oder mehrere beliebige BDOS-Funktionen zu erweitern, zu modifizieren und sogar neue Funktionen zu erzeugen.

Bekanntermaßen können CP/M-Programme die Systemfunktionen (BDOS-Funktionen) nutzen, indem sie mit festgelegten Parametern in den Prozessorregistern einen Unterprogrammaufruf zur Speicheradresse 0005h durchführen (CALL BDOS). Ist man jedoch aus irgendwelchen Gründen mit den vorhandenen Funktionen nicht zufrieden, besteht bei CP/M-Plus nunmehr auch offiziell die Möglichkeit, das Betriebssystem mit eigenen Routinen zu ändern oder zu erweitern. Dazu hat man sogenannte RSX-Module (Resident System Extensions, auf deutsch: feste Systemerweiterungen) eingeführt, die durch einen einfachen Trick so an das BDOS 'angeschlossen' werden, daß sie von einem Anwenderprogramm aus nicht mehr als eigenständige Module zu erkennen sind:

An der Adresse 0005h, also dort, wo ein BDOS-Call als erstes landet, steht normalerweise ein direkter Sprung zum Beginn des BDOS. Beim Laden eines RSX-Moduls wird dieser jedoch so verändert, daß der Aufruf anschließend am Anfang der RSX aufläuft. Der 'umgeleitete Sprung' ermöglicht somit dem RSX-Modul, alle künftigen BDOS-Aufrufe abzufangen und entsprechend den Vorgaben durch den Programmierer zu beeinflussen.

Um den Ladevorgang für RSX-Module möglichst einfach zu gestalten, bindet man sie an die Programme an, für die man sie benötigt. Dies geschieht mit dem CP/M-Dienstprogramm GENCOM, wobei sich jedem Programm eine oder mehrere RSX anhängen lassen. Die RSX werden mit dem Programm zusammen geladen und sind danach aktiv. Ob bis zum nächsten Warmstart oder 'für immer' (bis zum nächsten Kaltstart), entscheidet ein Flag in der RSX, dazu später mehr.

Die Idee zu meiner ersten RSX kam mir durch ein Spielprogramm. In diesem Programm gab es erstens keine Möglichkeit, den Spielverlauf zu unterbrechen und nach einer Pause mit dem Spiel fortzufahren, zweitens ertönte bei jedem nur erdenklichen Spielzustand der 'Beeper' meines Terminals.

Die notwendigen Routinen waren schnell geschrieben: 37 Zeilen Assembleranweisungen fangen alle BDOS-Aufrufe für die Funktionen 'Console Output' (BDOS-Funktion 2 und 6) sowie 'Console Input' (BDOS-Funktion 1 und 6) ab. Will das Programm ein Zeichen zur Konsole ausgeben, so vergleicht die RSX das Zeichen mit dem 'Klingel'-Code. Bei Übereinstimmung blockiert das RSX-Modul die Weitergabe dieses Zeichens und kehrt zum aufrufenden Programm zurück. Andernfalls wird das Zeichen an das BDOS weitergeleitet.

Fordert das Programm eine Konsoleingabe an, führt die RSX ihrerseits den zugehörigen BDOS-Aufruf durch und vergleicht alle vom BDOS zurückgegebenen Zeichen mit dem 'Pausezeichen' (zum Beispiel 'ESC'). Beim Empfang dieses Zeichens verzweigt die RSX zu einer 'Direct Console I/O' (BDOS-Funktion 6) mit 0FDh in Register E. Das BDOS wartet dann auf den nächsten Tastendruck, bis dahin wird das aufrufende Programm unterbrochen.

Durch die modulare Struktur des RSX-Konzepts ist es möglich, eine einmal definierte RSX an beliebige Programme anzubinden, sei es um ein bestehendes Programm zu erweitern oder die um Hauptfunktionen des Betriebssystems zu verändern.

RSX – wo?

Die an ein Programm angebundenen RSX werden also mit dem Programm zusammen in den Speicher geladen. Es dürfte einleuchten, daß das nicht alles gewesen sein kann. Denn schließlich brauchen die RSX ein sicheres Plätzchen, wo sie nicht überschrieben werden können, aber auch das auszuführende Programm nicht stören. Bei CP/M kann das nur das obere Ende der 'Transient Program Area' (TPA) sein.

Der Sprungbefehl auf Adresse 0005h dient ja nicht nur dazu, von einem Anwenderprogramm aus die BDOS-Funktionen aufzurufen. Sein Operand markiert zugleich das Ende des nutzbaren Speichers: Der Sprung führt zum ersten Byte des BDOS, ab hier ist keine TPA mehr.

So betrachtet erhält der 'umgeleitete Sprung' noch eine ganz andere Bedeutung. Auf diese Weise ist eine RSX mit ihrer Einbindung in den Sprung von Adresse 0005h zum BDOS automatisch überschreibgeschützt. Fragt ein Anwenderprogramm den Operanden ab (um das obere TPA-Ende zu bestimmen), findet es die Anfangsadresse der RSX – und 'denkt', es wäre die des BDOS.

Die Kehrseite der Medaille: Der Speicherplatz für RSX geht der TPA verloren. Aber normalerweise braucht man, was die Zahl der RSX-Bytes anbelangt, deshalb nicht gleich knausrig zu werden. Eine RSX ist meist konzentrierter Programmcode, und da ist 1 KByte schon ziemlich viel. Oft gibt es auch gar keinen Grund 'Platzangst' zu bekommen, beispielsweise wenn man eine Spezial-RSX für ein Programm erstellt, das die TPA gar nicht ausnutzt. Außerdem kann die Größe einer RSX ohnehin nur ganzzahlige Vielfache von 256 (100h) betragen.

Damit wäre geklärt, wo sich die RSX im Speicher aufzuhalten

haben, aber noch nicht, wie sie dorthin gelangen. Diese Aufgabe fällt dem CP/M-Programmlader zu, der praktisch eine RSX für den 'Command Control Processor' (CCP) darstellt. Sie erinnern sich: Bei CP/M-Plus liegt der CCP wie ein Anwenderprogramm ab 100h im Speicher. Um nun nicht mit dem zu ladenden Programm ins Gehege zu kommen, muß der Programmabschnitt, der das Laden übernimmt, aus dem CCP ausgelagert werden. Auch dieses Modul landet im Speicherbereich unterhalb des BDOS, und zwar immer als erstes (vom BDOS aus gesehen).

Erkennt nun der Lader an einem speziellen Vorspann, daß ein Programm eine oder mehrere RSX mit sich führt, so trennt er diese einzeln vom übrigen Programmcode, verschiebt sie ans Ende der TPA und 'verbiegt' den Sprungbefehl auf Adresse 0005h wie beschrieben. Dabei initialisiert er noch einige Bytes im sogenannten RSX-Präfix. Die erste RSX kommt direkt unter den CP/M-Lader zu liegen, alle weiteren unter die jeweils zuletzt geladene RSX. Abschließend wird der Programmcode richtig plaziert (durch den Vorspann beginnt er nicht mehr bei 100h sondern bei 200h) und das Programm gestartet.

RSX – wie?

Das RSX-Präfix (oft auch englisch mit 'e' geschrieben) sind die ersten 27 Bytes eines RSX-Moduls. Hier stehen diverse Informationen, die sowohl der Lader als auch die RSX selbst braucht. Einige davon muß der Programmierer festlegen, die anderen werden beim Laden gesetzt. Auch der CP/M-Programmlader besitzt dieses Präfix, das im einzelnen folgende Felder umfaßt:

SERIAL NUMBER
In diese sechs Platzhalter-Bytes kopiert der Lader die Seriennummer des vorliegenden CP/M. Dadurch enthalten die sechs Bytes vor dem Systemeinsprung, dessen Adresse bei 0006h steht, immer die Seriennummer, gleich, ob RSX vorhanden sind oder nicht. Dies dient der Kompatibilität mit den älteren CP/M-Versionen.

START
enthält die Sprunganweisung zum Beginn des RSX-Programmcodes. Diesen Befehl muß der Programmierer definieren.

NEXT
Bei diesem Sprungbefehl – er führt zum Beginn des nächsten RSX-Moduls oder zum BDOS – muß der Programmierer lediglich das Opcode-Byte setzen (0C3h), die Zieladresse bestimmt der Lader.

PREVIOUS
zeigt auf den Anfang des vorherigen RSX-Moduls oder auf die Adresse 0005h (wird vom Lader eingesetzt).

REMOVE
ist ein Flag, das der Programmierer setzen muß. Hat dieses Byte den Wert 0FFh, wird die betreffende RSX beim nächsten Aufruf des Programmladers über die BDOS-Funktion 59 (Load Overlay) aus dem System entfernt, zum Beispiel beim Warmstart. Ist das Flag gelöscht (00h), bleibt die RSX im System, bis entweder ein Kaltstart erfolgt oder die RSX selbst dieses Flag setzt.

NONBANKED
Noch ein Byte für den Programmierer. Mit diesem Flag legt er fest, ob die RSX immer geladen werden soll (00h) oder nur dann, wenn das vorliegende CP/M-Plus ein 'nonbanked system' ist (0FFh). So kann man zum Beispiel die in einem 'kleinen' System fehlenden BDOS-Funktionen mit einer RSX nachrüsten, die aber nur dann installiert wird, wenn es wirklich erforderlich ist.

NAME
Hier ist der Name des RSX-Moduls zu hinterlegen (maximal acht Zeichen, linksbündig, bei Bedarf mit Spaces aufzufüllen).

LOADER
markiert diejenige RSX, die direkt unterhalb des BDOS (0FFh) liegt. Dies ist immer der CP/M-Programmlader, bei allen übrigen RSX-Modulen wird das LOADER-Flag automatisch gelöscht (00h).

RESERVED
Für welchen Zweck, ist unbekannt – jedenfalls sind hier zwei Bytes freigehalten.

Wie zu Beginn erläutert, landen BDOS-Aufrufe zunächst in dem (den) RSX-Modul(en), ehe sie das BDOS erreichen. Diese als 'Preprocessing' bekannte Verfahrensweise erlaubt nicht nur, nicht vorhandene BDOS-Funktionen hinzuzufügen, sondern auch bestehende zu verändern oder völlig neu zu definieren. Im letzten Fall braucht man lediglich die zugehörigen RSX-Routinen mit einem Return-Befehl enden zu lassen statt mit einem Sprung zur nächsten RSX beziehungsweise zum BDOS.

Zumeist benutzt man RSX, um das BDOS zu manipulieren. Dazu muß eine RSX die übergebene BDOS-Funktionsnummer prüfen. Fühlt sie sich nicht angesprochen, so wird der 'CALL 0005h' kann unter Umständen zu einem bildschönen 'Hänger' führen, weil die RSX dann ja erneut durchlaufen wird.

Außerdem sollten RSX einen eigenen Stack einrichten, wie es auch das BDOS tut. Im Vertrauen darauf, daß der Stack ab dem BDOS-Aufruf nicht weiter belastet wird, ist er in vielen Anwenderprogrammen relativ knapp bemessen. Selbstverständlich sind die ursprünglichen Parameter vor Verlassen der RSX wiederherzustellen.

```
SERIAL:   db   0,0,0,0,0,0    ; Hierher kopiert der Lader die
                              ; BDOS-Seriennummer
START:    jmp  RSXstart       ; Sprung zum Beginn des RSX-Codes
NEXT:     jmp  $-$            ; Sprung zum START des folgenden
                              ; Moduls (Adresse wird vom Lader
                              ; eingesetzt)
PREV:     dw   0              ; Platzhalter fuer die Adresse des
                              ; vorhergehenden RSX-Moduls
REMOVE:   db   0ffh           ; 'Remove'-Merker
NONBANK:  db   0              ; 'Nonbank'-Merker
NAME:     db   'RSX Name'     ; Name des RSX-Modules
LOADER:   db   0              ; 0FFh markiert den Lader
RESERVE:  dw   0              ; reserviert

RSXstart:                     ; Hier folgen die RSX-Routinen
          .
          .
          .
```

Mit diesen 27 Bytes beginnt jede RSX.

BDOS-Aufruf durch einen Sprung zum NEXT-Feld im Präfix weitergereicht.

Wenn ein RSX-Modul selbst einen BDOS-Aufruf durchführen muß, ist auch dafür das NEXT-Feld zu verwenden. Nur so ist gewährleistet, daß das BDOS (eventuell nach Durchlaufen weiterer RSX) in jedem Fall erreicht wird. Der übliche

RSX – wozu?

Anfangs mag man meinen, der RSX-Technik seien schnell Grenzen gesetzt. Dieser Eindruck verliert sich in der Regel jedoch schnell, wenn man die ersten Gehversuche auf diesem ungewohnten Boden hinter sich hat. Tatsächlich sind die Möglichkeiten, dem BDOS mit RSX 'auf die Sprünge zu helfen', schier unbegrenzt. Im folgenden sind einige wenige Beispiele aufgeführt:

Eine RSX kann Parameter verändern, ehe der ursprüngliche BDOS-Aufruf weitergegeben wird. In diesem Fall ist die letzte Anweisung innerhalb des RSX-Moduls ein 'JMP NEXT'.

Sie kann einen BDOS-Aufruf durch einen oder mehrere andere BDOS-Aufrufe ersetzen und danach entweder per RET-Befehl zum aufrufenden Programm zurückkehren oder den Aufruf durch einen 'JMP NEXT' weitergeben.

Oder sie kann eine BDOS-Funktion komplett innerhalb der RSX nachbilden, ohne irgendeine BDOS-Funktion zu benötigen. So kann man zum Beispiel direkt auf spezielle Hardware zugreifen, die nicht ins CP/M eingebunden ist.

Der letztgenannte Fall reißt einen Aspekt an, von dem bisher noch gar nicht die Rede war: Per RSX kann man auch vorübergehende BIOS-Änderungen realisieren, etwa um das CP/M-Plus-BIOS kompatibel zum BIOS von CP/M 2.2 zu machen. Dann muß die RSX eine eigene BIOS-Sprungleiste enthalten und diese beim ersten Aufruf in das System 'einklinken', indem sie den Warmstartsprung auf Adresse 0000h 'verbiegt'. Bei der Gelegenheit kann

die RSX auch gleich die erforderlichen Initialisierungen vornehmen. Danach sind die zugehörigen Routinen lahmzulegen. Sofern die RSX nicht auch BDOS-Funktionen beeinflussen soll, braucht sie dazu bloß das START-Feld mit einem 'JMP NEXT' zu überschreiben. Eins darf man bei der ganzen Aktion allerdings nicht vergessen – den ursprünglichen BDOS-Aufruf weiterzugeben.

Nach der ganzen Theorie zum Abschluß noch ein praktisches Beispiel, das zum einen sehr genau die Struktur eines RSX-Moduls zeigt und außerdem eine der Unverträglichkeiten zwischen CP/M 2.2 und CP/M-Plus beseitigt:

Unter CP/M 2.2 ist es zulässig, daß das Namensfeld in einem FCB Fragezeichen enthalten darf, auch wenn die aufzurufende BDOS-Funktion die Funktion 15 ist ('Open File'). In dem Fall wird das BDOS die erste Datei im Directory eröffnen, deren Name mit dem angegebenen übereinstimmt. Wenn nun ein Programm, das dies ausnutzt, unter CP/M-Plus gestartet wird, gibt das BDOS die Fehlermeldung '? in filename' aus und kehrt mit dem Parameter für 'file not found' zum aufrufenden Programm zurück.

Mit dem Beispiel-RSX-Modul kann man dieses unerwünschte Verhalten ausschalten. Es modifiziert die Systemfunktion 'Open File' wie folgt: Bei einem BDOS-Aufruf mit der Funktionsnummer 15 wird der übergebene FCB auf '?' im Namen überprüft. Enthält der Name kein '?', so wird der Aufruf unverändert an das BDOS weitergeleitet. Andernfalls wandelt die RSX den Open-Aufruf in einen Aufruf der BDOS-Funktion 17 um ('Search First'). Findet diese Funktion keine passende Datei, gibt die RSX die Meldung 'file not found' an das aufrufende Programm zurück. Ist Funktion 17 jedoch erfolgreich, überträgt die RSX den gefundenen Dateinamen anstelle des ursprünglichen in den FCB und ruft danach die Funktion 'Open File' auf. Das BDOS öffnet nun wie gewünscht die Datei und kehrt zum aufrufenden Programm zurück.

Natürlich muß das neuerstellte RSX-Modul auch getestet werden, um seine Funktionstüchtigkeit sicherzustellen. Hierzu dient das als zweites abgedruckte Testprogramm. Es fordert die Eingabe eines Dateinamens an, erstellt mittels der BDOS-Funktion 'Parse Filename' (Nummer 152) einen entsprechenden FCB und ruft dann Funktion 'Open File' auf. Ist das RSX-Modul eingebunden und man gibt einen Mehrfach-Dateinamen ein (ambiguous filename), dann wird die erste Datei aus dem Directory eröffnet, auf die die Beschreibung paßt. Ohne RSX erscheint wie üblich eine Fehlermeldung.

Werdegang

Damit man sich eines RSX-Moduls erfreuen kann, muß es in 'maschinenlesbarer' Form vorliegen. Die ersten Schritte dahin sind dieselben wie bei jedem Programm: Man erstellt den Quelltext und läßt ihn übersetzen. Ein geeigneter Assembler ist zum Beispiel der bei CP/M-Plus mitgelieferte RMAC-Makroassembler, man kann aber auch andere verwenden, sofern sie den Objekt-Code in einer REL-Datei ablegen.

Die Objekt-Datei wird dann mit dem Linker von Digital Research (LINK) weiterbehandelt. Dieser ist mit der Option [OP] aufzurufen und erzeugt aus der REL- eine PRL-Datei (Page Relocatable). Der Unterschied zwischen beiden Arten verschiebbaren Objekt-Codes besteht darin, daß man die Anfangsadresse für den lauffähigen Code bei einer REL-Datei frei wählen kann, bei einer PRL-Datei dagegen muß das niederwertige Byte der Adresse Null sein. Das PRL-Format hat gegenüber dem REL-Format den Vorteil, daß der Aufwand für das Verschiebeprogramm sehr gering ist. Das PRL-Format ist übrigens der Grund dafür, daß die Größe einer RSX nur in ganzen 256-Byte-Blöcken 'gehandelt' wird.

Der vorletzte Schritt ist, die PRL-Datei in eine RSX-Datei umzubenennen. Ohne dies führt der letzte Schritt, das Anbinden der RSX an ein Programm mit Hilfe von GENCOM, nicht zum Erfolg.

Literatur

CP/M Plus Programmer's Guide, Digital Research, 1982

Garry M. Silvey, Resident System Extensions, Dr. Dobbs Journal, Juli 84

```
                                        ***

; Open File RSX
;
; Dieses RSX-Modul fängt den BDOS-Aufruf 'Open File' ab.
; Wenn im vorgegebenen FCB '?' auftreten, so wird vor 'Open File'
; die Funktion 'Search First' eingeschoben, damit kein BDOS-
; Error auftritt, und mit dem gefundenen Dateinamen die Open-
; Funktion ausgeführt. Wenn nicht, wird der Aufruf direkt an
; das BDOS weitergeleitet.
;

TRUE    equ  0ffffh
FALSE   equ  NOT TRUE

PRINTSTR equ  09              ; Print String Funktion
OPENFILE equ  15              ; Open File Funktion
SEARCHF  equ  17              ; Search First Funktion
SCB      equ  49              ; Get/Set System Control Block
CALLRSX  equ  60              ; Call RSX

DEBUG   equ  TRUE             ; Debug-Modus liefert mehr Meldungen
                              ; für Tests
Z80     equ  FALSE            ; 8080/8085-Code wird generiert


; Beginn RSX-Prefix

        db   0,0,0,0,0,0      ; Seriennummer
        jmp  START            ; Beginn des RSX-Codes
NEXT    jmp  $-$              ; Sprunganweisung
PREV    dw   0                ; Adresse vorheriges Modul
REMOVE  db   0ffh             ; lösche bei Warmstart
NONBANK db   0                ; lade immer
RSXNAME db   'OPEN    '       ; Name des Modules (8 Stellen!)
        db   0,0,0            ; benutzt von Lader und BDOS

; Beginn RSX-Code

START   mov  a,c              ; prüfe BDOS Funktion
        cpi  OPENFILE         ; ist dies ein 'Open File' Aufruf ?
        jz   GEFANGEN         ; wenn ja, dann tu's

; wenn DEBUG eingeschaltet, dann prüfe zusätzlich, ob dies ein
; BDOS-Aufruf 'Call RSX' ist. Wenn ja, dann kehre mit Akku =0
; zum rufenden Programm zurück, und teile mit, daß das RSX-Modul
; geladen und aktiv ist.

if DEBUG

        cpi  CALLRSX          ; ist es ein Aufruf für diese RSX ?
        jnz  NEXT             ; nein, weiter zur nächsten RSX
                              ; oder zum BDOS
        mov  a,d              ; liegt ein Testprogramm-Aufruf an ?
        cpi  0ffh
        jnz  NEXT
        cmp  e                ; Register DE = 0ffffh ?
        jnz  NEXT             ; nein, gebe Aufruf weiter
        xra  a                ; andernfalls setze Register A = 0
        ret                   ; zurück zum rufenden Programm

else
        jmp  NEXT             ; zur nächsten RSX oder BDOS wenn
                              ; keine 'Open File'-Funktion

endif

GEFANGEN lxi  h,0
        dad  sp               ; sichere aktuellen Stackpointer
        shld USERSTACK
        lxi  sp,RSXSTACK      ; richte lokalen Stack ein
        push b                ; sichere Benutzerregister
        push d

; überprüfe Dateiname auf '?'

        mvi  b,11             ; max. Zeichenanzahl im Dateinamen
        inx  d                ; setze Zeiger auf 1. Namenszeichen
LOOP    ldax d
        cpi  '?'              ; ist es ein '?' ?
        jz   ERSETZE          ; '?' gefunden, führe RSX aus
        inx  d                ; nein, nächstes Zeichen auch nicht?
        dcr  b                ; stelle Zeichenzähler nach
        jnz  LOOP

; wenn das Programm hier angekommen ist, so befinden sich keine
; '?' im Dateinamen und der BDOS-Aufruf wird unverändert
; weitergeleitet

        pop  d                ; Registerinhalte wiederherstellen
        pop  b
```

```
        lhld USERSTACK
        sphl                    ; ebenso den alten Stackpointer
        jmp  NEXT               ; und weiter zu BDOS

ERSETZE: ; Hier wird die 'Search First'-Funktion eingeleitet,
        ; der FCB mit dem erstgefundenen Dateinamen gesetzt und
        ; abschließend 'Open File' aufgerufen.
        ; Wenn der 'Search First' ein Fehler meldet, dann meldet
        ; sich die RSX mit einer entsprechenden Fehlermeldung an
        ; das aufrufende Programm zurück.

if DEBUG

        lxi  d,MELDUNG          ; melde daß ein '?' gefunden wurde
        mvi  c,PRINTSTR         ; 'Call BDOS' über NEXT !!
        call NEXT

endif

        pop  d                  ; lade Adresse des FCB
        push d                  ; und sichere erneut
        mvi  c,SEARCHF          ; bereite Funktion vor
        call NEXT               ; und rufe BDOS
        cpi  0ffh               ; nicht gefunden ?
        jnz  GEFUNDEN           ; gefunden, so fahre fort

        ; Hierher kommt das Programm nur, wenn 'Search First'
        ; nicht erfolgreich war. In diesem Fall unterbleibt der
        ; Open-Aufruf, da immer noch '?' im Dateinamen stehen,
        ; und wir kehren mit einer entsprechenden Fehlermeldung
        ; (Register A bzw. HL) zum rufenden Programm zurück.

        pop  d                  ; Registerinhalte wiederherstellen
        pop  b
        lhld USERSTACK          ; ebenso den alten Stackpointer
        sphl
        ret                     ; zurück zum Programm

GEFUNDEN ; 'Search...' war erfolgreich, ersetze alten FCB durch
        ; den der gefundenen Datei und rufe 'Open File' auf.
        ; Die 'Search First'-Funktion legt im aktuellen DMA-
        ; Bereich den Directory-Sektor ab (128 Bytes), in dem
        ; die gefundene Datei steht, und teilt in Register A
        ; mit, welcher dieser 4 Einträge dazugehört (0...3).
        ; Der Wert in Register A mit 32 multipliziert ist der
        ; Offset des gültigen Eintrags bezogen auf den Anfang
        ; des Sektors (32 = Länge eines Directory-Eintrages).

        add  a
        add  a
        add  a
        add  a
        add  a
        push a                  ; sichere Offset

        ; Um den gefundenen Namen laden zu können, braucht die
        ; RSX die aktuelle DMA-Adresse, die im SCB steht.

        mvi  c,SCB              ; 'Get/Set SCB'-Funktion
        lxi  d,SCBPB            ; SCB Parameter Block Adresse
        call NEXT               ; BDOS Aufruf

        ; Register HL enthält nun aktuelle DMA-Adresse

        pop  a                  ; lade Offsetwert wieder
        mov  c,a
        mvi  b,0                ; setze Offset in Register BC
        dad  b                  ; HL zeigt nun auf den gefundenen FCB
        inx  h                  ; setze auf 1.Zeichen des Dateinamens
        pop  d                  ; lade ursprüngliche FCB-Adresse
        push d
        inx  d                  ; setze auf 1. Zeichen Ur-Dateiname

        ; Kopiere den Namen der gefundenen Datei in den FCB des
        ; aufrufenden Programms (Ur-FCB).

if Z80
        ; Mit einer Z80-CPU geht das per LDIR, anderfalls
        ; (8080/8085) muß dieser Befehl simuliert werden.

        lxi  b,11               ; Anzahl Zeichen in Dateiname
        dw   0b0edh             ; Z80 Ldir-Befehl

else
        mvi  b,11
LDIR    mov  a,m
        inx  h
        stax d
        inx  d
        dcr  b
        jnz  LDIR

endif

        ; Der FCB des aufrufenden Programms ist nun bereit
        ; für den 'Open File'-Aufruf.

        pop  d                  ; Alte Register wiederherstellen
        pop  b                  ; (Register C enthält dann wieder
                                ; die Funktionsnummer für Open)
        lhld USERSTACK
        sphl                    ; Alten Stackpointer wiederherstellen
        jmp  NEXT

        ; Das BDOS wird jetzt ganz normal die Datei öffnen.


SCBPB:  ; SCB Parameter Block: Eingerichtet um die aktuelle DMA-
        ; Adresse zu speichern (Adresse im SCB mit Offset 03ch)
OFFSET      db   03ch
WERT        db   0          ; 'Get'-Funktion

USERSTACK   dw   0          ; Ablage für alten Stackpointer
            ds   10         ; Bereich für lokalen RSX-Stack
RSXSTACK    equ  $          ; RSX-Stack Basis

        ; Wenn DEBUG eingeschaltet ist, so wird folgende Meldung
        ; generiert

if DEBUG

MELDUNG db   '? im Dateinamen, abgefangen von )OPEN.RSX( ...'
        db   0dh,0ah,'$'

endif

        end
```

Eine einfache RSX, um die 'Open File'-BDOS-Funktion von CP/M-Plus auf den Stand von CP/M 2.2 'zurückzuwerfen'.

```
                                        ***

; Testprogramm für die 'Open File RSX'
;
; Das Programm fordert die Eingabe eines Dateinamens an, erstellt
; den entsprechenden FCB mit Hilfe der BDOS-Funktion 152 ('Parse
; Filename') und versucht dann, die Datei zu eröffnen.
; Enthält der Dateiname '?', so bricht das BDOS normalerweise
; (ohne RSX) mit einer Fehlermeldung ab. Mit der 'Open File RSX'
; erfolgt das öffnen der Datei wie unter CP/M 2.2 und die
; Fehlermeldung bleibt aus.

TRUE    equ  0ffffh
FALSE   equ  not TRUE
BDOS    equ  0005h          ; BDOS Einsprungadresse
CR      equ  13             ; Carriage Return
LF      equ  10             ; Line Feed

; BDOS-Funktionen

PRINTSTR equ  09            ; Print String Funktion
RBUFF   equ  10             ; Read Console Buffer
OPENFILE equ  15            ; Open File Funktion
SEARCHF equ  17             ; Search First Funktion
SCB     equ  49             ; Get/Set SCB Funktion
CALLRSX equ  60             ; Call RSX Funktion
PARSE   equ  152            ; Parse Filename

DEBUG   equ  TRUE

        org  0100h

START   ; gebe Meldung aus
        lxi  d,MELDUNG
        mvi  c,PRINTSTR
        Call BDOS

if DEBUG

        ; Sieh nach, ob die RSX überhaupt eingebunden ist.
        ; Dazu dient die BDOS-Funktion 60 ('Call RSX'), die beim
```

```
        ; Aufruf in Register DE die Adresse eines RSX-Parameter-
        ; blocks erhält. Die 'Open File RSX' braucht jedoch keine
        ; Parameter, daher kann man Register DE mit 0ffffh laden,
        ; um diesen RSX-Aufruf von eventuellen Aufrufen anderer
        ; RSX zu unterscheiden.

        mvi  c,CALLRSX
        lxi  d,0ffffh
        call BDOS
        cpi  0              ; Ist eine RSX aktiv ?
        jz   RSXAKTIV       ; Ja!
        mvi  c,PRINTSTR
        lxi  d,NEINTEXT     ; Melde daß keine RSX da
        call BDOS

RSXAKTIV mvi  c,PRINTSTR    ; Melde daß RSX vorhanden
        lxi  d,JATEXT
        call BDOS


endif

        mvi  c,PRINTSTR     ; Fordere Dateinamen an
        lxi  d,PROMPT
        call BDOS

        lxi  d,EINPUFFER    ; Lege Eingabe in Puffer ab
        mvi  c,RBUFF
        call BDOS

        lxi  RETURN         ; Neue Zeile
        mvi  c,PRINTSTR
        call BDOS

        lxi  d,PFCB         ; Setze Parse Filename Control Block
                            ; und übergebe Dateiname
        mvi  c,PARSE
        call BDOS

        lxi  d,FCB          ; Dateiname nun in FCB, öffne Datei
        mvi  c,OPENFILE
        call BDOS
        cpi  0ffh           ; Fehlermeldung von BDOS ?
        jnz  GEFUNDEN       ; Wenn ja, Fehlermeldung ausgeben
                            ;                       und Ende
        lxi  d,FEHLER       ; Meldung Fehler aufgetreten
        mvi  c,PRINTSTR
        call BDOS

EXIT    mvi  c,0            ; Warmboot
        jmp  BDOS


GEFUNDEN lxi  d,OKTEXT      ; Melde gefundene Datei
        mvi  c,PRINTSTR
        call BDOS
        lxi  d,FCB+12       ; setze DE auf Ende des Dateinamens
        mvi  a,'$'          ; speichere dort ein '$' ab (String
        stax d              ;                       Delimiter)
        lxi  d,FCB+1
        mvi  c,PRINTSTR
        call BDOS           ; gebe Dateinamen aus
        lxi  d,RETURN       ; Neue Zeile
        mvi  c,PRINTSTR
        call BDOS
        jmp  BDOS           ; Rückkehr zum CCP


PFCB    ; Parse Filename Control Block
        dw   EINPUFFER+2    ; Adresse Eingabepuffer
        dw   FCB

EINPUFFER    ; Console Input Eingabepuffer
        db   14,0           ; Anzahl Zeichen (zum späteren Lesen)
        ds   14             ; Länge Eingabepuffer
FCB     ds   36             ; FCB Bereich

        ; Meldetexte

FEHLER  db   'Datei nicht gefunden',CR,LF,'$'
MELDUNG db   ')Open File RSX( Test Programm ... ',CR,LF,'$'
OKTEXT  db   'Name der gefundenen Datei : $'
RETURN  db   CR,LF,'$'
JATEXT  db   'RSX eingebunden.',CR,LF,'$'
PROMPT  db   'Dateinamen eingeben : $'
NEINTEXT db  07,'ACHTUNG!! Kein $'

        end
```

Ein Testprogramm für die Beispiel-RSX.

c't-Klangcomputer

Polyphoner Synthesizer zum Selbstbau

- bis zu acht Klangspeicher
- Naturklang-Verarbeitung
- Sound-Editing in Hi-Res-Grafik
- Klangspeicherung auf Diskette
- Menügeführte Bedienung
- c't-Serie ab Heft 12/84

Zum Aufbau des c't-Klangcomputers werden ein betriebsfähiges Apple-II-Motherboard mit Floppy-Controller und mindestens ein Laufwerk benötigt.

Leerplatinen:

ADS
Analog-/Digital-Sampler und Vorverstärker zur Digitalisierung beliebiger Audiosignale, beide Platinen zusammen ..... **38 DM**

KBI
Keyboard-Interface zum Anschluß der Klaviatur, mit zusätzlichem Midi-Ausgang ..... **39 DM**

KBC
Keyboard-Control-Karte zur Betriebsarten-Einstellung und Midi-Interface-Bedienung ..... **22 DM**

KBB/KBE
Klaviatur-Basiskarten und -Erweiterungskarte, Abfragelogik für beliebige Tastaturen mit 61 Tasten (ein Schließer pro Taste), Satz aus 1x KBB und 3x KBE ..... **85 DM**

PCS
Parameter-Control-System zur Ansteuerung der Klangspeicher (für alle VCOs mit 1V/Oktave-Charakteristik geeignet) ..... **42 DM**

Voice RAM
Klangspeicher zur tastaturgesteuerten Wiedergabe eines Klangs (Maximalausbau: 8 Stimmen = 8 Voice-RAM-Karten). **49 DM**
Voice-RAM-Satz, bestehend aus acht Karten ..... **369 DM**

Programmdiskette
ADS-Steuerung, Klang-Editor, Synthesizer-Betriebssystem, Demo-Klänge ..... **25 DM**

Kompletter Satz
Maximalausbau (ADS-Vorverstärker, ADS-Slotkarte, KBI-Slotkarte, KBC, KBB, 3x KBE, PCS, 8x Voice RAM, Programmdiskette) **598 DM**

Auch unabhängig vom Klangcomputer einsetzbar:

DSM
Digital Sound Module zum anschlagdynamischen „Abspielen" von digitalisierten (Natur-)Klängen aus EPROMs, besonders geeignet für Schlagzeug ..... **15 DM**

Sound Samples
Hochwertige Studioaufnahmen für das DSM

Volume 1:

Doppeltom	Snare, heavy	Snare, Hi4	Rimshot
Timpani	Snare, medium	Snare, Hi5	Claprx
Tom1	Conga	BD, heavy	HH, closed
Tom2	Snare, Hi1	BD, mid	HH, open
Tom3	Snare, Hi2	BD, High	Snare, gated
Tom4	Snare, Hi3	Cowbell	BD, gated

Volume 2:

Simmons, hitom	Strums	HH, closed	Glas
Simmons, midtom	Crash	HH, open	Snare, lutz, m
Simmons, lotom	Ride	Pauke	Rimshot
Clap	Bassdrum	Snare, reggae	Timbale

Diskette Vol. 1 (Apple II) **35 DM** Kassette (Vol. 1 + Vol. 2) (SuperTape) **69 DM**
Diskette Vol. 2 (Apple II) **35 DM** EPROM (Typen 2716...27128), je Instr. **25 DM**

Eine Kurzbeschreibung der verschiedenen Klänge erhalten Sie gegen Zusendung eines rückadressierten Freiumschlags.

So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck über die Bestellsumme zuzüglich DM 3,— (für Porto und Verpackung) bei oder überweisen Sie den Betrag auf eines unserer Konten.

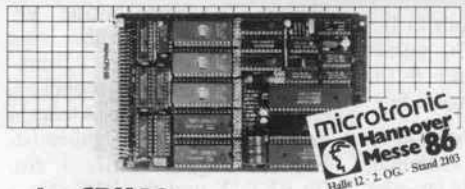
Schecks werden erst bei Lieferung eingelöst. Wir empfehlen deshalb diesen Zahlungsweg, da in Einzelfällen längere Lieferzeiten auftreten können.

Bankverbindungen: Postscheckamt Hannover, Kt.-Nr. 93 05-308
Kreissparkasse Hannover, Kt.-Nr. 000-019968 (BLZ 250 502 99)

Ihre Bestellung richten Sie bitte an:

Verlag Heinz Heise GmbH
Postfach 61 04 07, 3000 Hannover 61

EINKARTEN-RECHNER



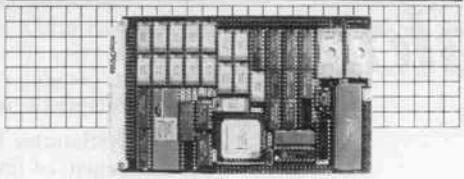
miroCPU80

- ECB-Steuerungs-CPU für den industriellen Einsatz
- 6 MHz Z80 B-CPU, max. 512 KByte adressierbar
- 5 × 28pol. EPROM/RAM-Fassungen, alle Typen einsetzbar
- 3 Serienschnittstellen mit Z80-SIO u. Z80-STI (2× V.24, 1× TTL), 8 bit Parallelport
- Echzeituhr, Timer, Power-Fail-Logik
- Batteriepufferung für CMOS-RAM und Uhr
- 4fach-Multilayer, Industriequalität

miro Datensysteme GmbH
Gifhorner Str. 28, 3300 Braunschweig
Telefon 05 31/3 27 91

miro datensysteme

16 BIT SLAVERECHNER



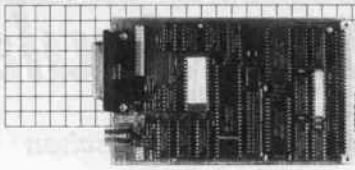
miroCPU186

- 16-Bit-Einkartenrechner, ECB-Interf. über FIFO
- 8 MHz 80186 CPU mit DMA-/Interrupt-Controller
- 128/512 KByte RAM, 2 28pol. EPROM-Steckplätze
- 2 prozedurf. Serienschnittstellen sync./async.
- 8087 Numeric-Prozessor u. Speichererweiterung auf 1 MByte über lokalen Bus anschließbar
- CP/M-86 auf jedem ECB-Rechner ohne Anpassung
- Turbo DOS-86 für Multiprozessor-Anwendungen

miro Datensysteme GmbH
Gifhorner Str. 28, 3300 Braunschweig
Telefon 05 31/3 27 91
Teletex 5 318 146 = miro, über Tx. 1 75 318 146

miro datensysteme

LOW-COST CRT-CONTROLLER



miroVIDEO*LC*

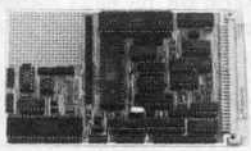
- Bildformate: 80 x 24, 40 x 24, 40 x 12, 26 x 8, Statuszeile beschreibbar und abschaltbar
- V. 24-Interface (110–19 200 Baud mit Handshake) oder ECB-Businterface wahlweise einstellbar
- Attribute: normal, dim, underline, blank, blink doppelte Höhe und/oder Breite, beliebig kombinierbar
- Zeichensätze: ASCII, Deutsch, Blockgraphik
- Emulationen: VT52 und Televideo
- Tastatur: 8 Bit parallel oder Matrix
- Ausgänge: BAS oder TTL Video, Bell

miro Datensysteme GmbH
Gifhorner Str. 28, 3300 Braunschweig
Telefon 05 31/3 27 91
Teletex 5 318 146 = miro, über Tx. 1 75 318 146

miro datensysteme

Katalog kostenlos

UNIO-1



Universal Input Output Karte

- Z80-PIO, -SIO, -STI + 3 Latch-Ports
- Zwei gepufferte Centronics-Schnittstellen
- Zwei V24/RS232-Schnittstellen
- Eine 20mA-Schnittstelle m. Optokoppler
- RS232-Spannungswandler optional
- Wrap-Feld, 4 Timer, 8 Interrupteingänge
- 16 bidirektionale I/O-Leitungen
- ECB-Bus-Anschluß, voll interruptfähig

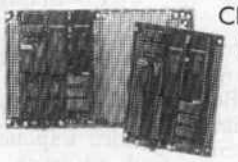
Fertiggerät	385.96	(440.- m. MwSt.)
Leerplatine + Handbuch	85.96	(98.- m. MwSt.)
Handbuch allein (wird angerechnet)	25.-	(m. MwSt.)

J.C. Lotter KG
D-6100 Darmstadt 11
Postfach 110622
Telefon: (06151) 26013
Telex: 4197298

CONITEC DATENSYSTEME

Katalog kostenlos

CEPAC-80



CMOS Einplatinen Allzweck Computer

- NSC800-CPU, Z80-kompatibel, 1 - 4 MHz
- 3 bidirektionale Ports (NSC810-RIOT)
- 16 Latches, optional mit 250 mA-Ausgängen
- 8 Signaleingänge, 5 Interrupteingänge
- 2 16-Bit-Timer; 1 14-Bit-**Watchdog-Timer**
- 128 Byte CMOS-RAM; 2 RAM/EPROM-Sockel
- **Stromverbrauch typ. 20 mA**
- ECB-Bus-Anschluß; Wrap-Feld optional
- Beschrieben in **c't** 1/84

Fertiggerät	217,54	(248.- m. MwSt.)
Leerplatine + Handbuch	42.98	(49.- m. MwSt.)
Wrap-Feld-Aufpreis	17.54	(20.- m. MwSt.)
Handbuch allein (wird angerechnet)	25.-	(m. MwSt.)

J.C. Lotter KG
D-6100 Darmstadt 11
Postfach 110622
Telefon: (06151) 26013
Telex: 4197298

CONITEC DATENSYSTEME

Floppy-Drive's* Streamer Festplatten

für IBM und kompatible Computer

Floppy-Drive:
Tandon TM 65 LL
2x40 Track-Slimline
nur DM 298,--
ab 3 St. DM 275,--

Festplatten-Einbauset:
bestehend aus:
1. Tandon Slimline-Platte 20 MB
2. Western Digital Controllen
3. Kabelsatz + Anleitung
nur DM 1.998,--
ab 3 St. DM 1.898,--

Tape-Streamer:
Teac MT-2 St/kit
20 MB-Slimline
nur DM 2.498,--
ab 3 St. DM 2.398,--

- Weitere auf Anfrage!
- Händleranfragen erwünscht!
- Wir liefern auch Plattensets und Streamer mit SASI-Bus!

Alle Preise inkl. 14 % MWSt.

Adressen:
Dorsch-Elektronic
Steinäcker 2
8551 Igensdorf
Tel. 09192/17 77

Mr. Bit - Computerladen
Mr. Bit COMPUTERLADEN
Forsthofstr. 35
8500 Nürnberg 40
Tel. 0911/47 27 77

Atari liefert den ST, doch ohne Software tut er weh!

Programmiersprachen:

Basic-M Interpreter (Philon)	**198,00 DM**	**Modula 2-ST** (TDI)	**349,50 DM**
Basic-M Compiler (Philon)	**399,00 DM**	**C-Compiler** o. Float. Point (GST)	**349,50 DM**
Pascal Compiler (Philon)	**519,00 DM**	**Assembler** (GST)	**149,00 DM**
C-Compiler incl. Floating Point	**598,00 DM**	**Assembler** (Kuma)	**198,00 DM**
Fortran Compiler (Philon)	**519,00 DM**	**Anwendungen:**	
Cobol Compiler (Philon)	**998,00 DM**	**K-Spreadsheet** (Kuma)	**198,00 DM**

city elektronik

Güntherstraße 75 · 4600 Dortmund 1 · Telefon (02 31) 57 22 84

Bitmanipulator

Bits setzen, abfragen, löschen in Spectrum-BASIC

Martin Müller

Der Umgang mit Bytes stellt dank der BASIC-Befehle PEEK und POKE kein Problem dar. Will man jedoch auf einzelne Bits in einem Byte zugreifen, wird's aufwendig, und die Rechenzeit steigt unter Umständen gewaltig an. Daher bietet es sich an, eine kurze Maschinensprachenroutine zu nutzen, die genauso komfortabel wie die Z80-Befehle BIT, RES und SET von BASIC aus Bits abfragt, löscht oder setzt.

Bitweise Operationen braucht man zum Beispiel, wenn man Attribute im Farbspeicher verändern oder Tastaturabfragen durchführen möchte, die nicht durchs Drücken einer weiteren Taste blockiert werden (geht meistens auch bei drei Tasten, wenn die gesuchte nicht gleichzeitig mit der einen Taste eine gemeinsame Matrixspalte und mit der anderen eine gemeinsame Matrixzeile besitzt). Aber auch wer von BASIC aus Interfaces steuern und abfragen möchte, kommt nicht ohne Bit-Technik aus.

Eine reine BASIC-Lösung ohne Maschinenprogrammhilfe ist als Funktion, also ohne Unterprogramm möglich. Leider interpretiert der Spectrum die Befehle 'AND', 'OR' und so weiter nicht bitweise wie etwa sein CPC-Konkurrent. Es muß also Ersatz her.

Gemütlich in BASIC

Hierfür kommt vor allem die zahlentheoretische Funktion INT zum Einsatz, die alle Nachkommastellen einer Fließkommazahl auf Null setzt. Um beispielsweise festzustellen, ob eine Zahl gerade oder ungerade ist (bei Bytes Abfrage von Bit 0) ist die Abfrage aufschlußreich:

$$x = 2 * \mathrm{INT}\ (x/2)$$

Als Antwort erhält man in bester ZX-Manier '1' für 'wahr', entspricht 'gerade', oder '0' für 'falsch', entspricht 'ungerade'.

Um ein bestimmtes Bit (das n-te) abzufragen, ist festzustellen, ob INT (x/2↑n) gerade ist oder nicht. Die BIT-Funktion sieht also folgendermaßen aus:

```
DEF FN B(x,n) =
INT (x/2↑n) <>
2 * INT ((INT (x/2↑n)/2)
```

Sie liefert (wegen <>) jetzt 0, falls das n-te Bit 0 ist und 1, falls das Bit gesetzt ist.
Die Funktionen RES und SET können nun diese BIT-Funktion mitbenutzen:

```
DEF FN R(x,n) =
x - FN B(x,n) * 2↑n

DEF FN S(x,n)=
x + ( NOT FN B(x,n)) * 2↑n
```

Die bei den Befehlen vorkommenden Exponentiationen 2↑n sind natürlich ausgesprochen zeitraubend und verlangsamen Programme meist beträchtlich. Man sollte sie daher durch ein Feld ersetzen mit Z(n+1)= 2↑n (der Feldindex muß leider um eins höher sein, da es kein nulltes Element gibt), das alle vorkommenden Potenzen von zwei enthält.

. . . und beschleunigt

Bei zeitkritischen Programmen ist aber manchmal auch das noch zu langsam. Hierfür ist die kleine Maschinenroutine gedacht, die ebenfalls über eine anwenderdefinierbare Funktion aufgerufen wird, wobei als erster Parameter die Auswahl der Operation zu übergeben ist. Für BIT trägt man hier '1' ein, für SET '2' und für RES '4'.

Als zweiter Parameter folgt das betroffene Bit, mit LSB (Least Significant Bit) = 0 und MSB (Most . . .) = 7.

Der dritte und letzte Parameter gibt den Operanden an, der das zu behandelnde Byte enthält. Es sollte sich daher um ein Ein-Byte-Integer (0 bis 255) handeln.

Die Operationen verändern die Operanden nicht. Will man beispielsweise ein Bit einer Variablen setzen, so sieht das so aus:

```
LET X= FN B(SET,4,X)
```

Das Maschinenprogramm besteht aus den vier 'Modulen'

```
Operations-
dekoder (160 - 280)
Reset Bit (300 - 350)
Set Bit (370 - 420)
Test Bit (440 - 500)
```

Wenn die jeweilige Routine abgearbeitet ist, wird das Ergebnis in das C-Register geladen (B-Register ist Null) und an das BASIC-Betriebssystem übergeben. Falls jedoch ein (noch) nicht definierter Wert dem Operationsdekoder übergeben wurde, enthält B den zweiten

```
  1 REM **********************
        *BITOPERATIONEN       *
        *BIT,RES,SET          *
        *  by M.Mueller       *
        **********************

  2 REM Der Maschinencode  ist
        Relokativ, es muessen
        nur die Zeilen 10,9000
        9010 und 9999 angepasst
        werden.
 10 DEF FN B(x,y,z)=USR 61000
 20  LET BIT=1
 30 LET SET=2
 40 LET RES=4
 50  REM Als Parameter muss die
        Operation (Bit-Test=1,
        setze Bit=2,loesche Bi
t       =4),das betroffene Bit
        und der Operand angege
-       ben werden.
 90  LET x=127: REM Operand
100 REM "Anwendung"
110 REM .....................
        1. Bit-Test
        .....................

120 PRINT FN B(BIT,2,X)
130  REM Ergibt 1,weil Bit 2
        (dezimal 4) gesetzt
        ist. Ergaebe 0 bei
        x=123.
200 REM .....................
        2. Setze Bit
        .....................

210 PRINT FN B(set,7,x)
220 REM Ergibt 255,weil Bit 7
        die Wertigkeit 128 hat.
        (127+128=255)
300 REM .....................
        3. Loesche Bit
        .....................

310 PRINT FN B(RES,3,x)
320 REM Ergibt 119, weil RESet
        ein Bit loescht, hier
        Bit 3 gleich dezimal 8.
        (127-8=119)
8999 STOP
9000 CLEAR 60999: RESTORE 9000:
DATA 243,221,42,11,92,221,126,4,
221,70,12,4,15,56,26,15,56,13,15
,48,34,175,47,23,16,253,221,166,
20,24,24,175,55,23,16,253,221,18
2,20,24,14,175,55,23,16,253,221,
166,20,40,4,62,1,24,0,79,251,201
,256
9010 LET a=61000
9020 READ x: IF x=256 THEN  GO T
O 1
9030 POKE a,x: LET a=a+1: GO TO
9020
9999 SAVE "BITMANI" LINE 9000: R
EM eventuell SAVE "BIT"CODE 6100
0,58
```

```
0010  ;               "**********************"
0020  ;               "* Bitwise Operations *"
0030  ;               "* By MPS 10/85       *"
0040  ;               "**********************"
0050  ;
0060  ;               "Parameter:"
0070  ;               "1.Operationsidentifier"
0080  ;               "   1 = Test Bit"
0090  ;               "   2 = Set Bit"
0100  ;               "   4 = Reset Bit"
0110  ;               "2.Betroffenes Bit 0-7"
0120  ;               "3.Operand"
0130  ;
0140          ORG     61000
0150          DI
0160          LD      IX,(23563)
0170  ;               "Operation dekodieren"
0180          LD      A,(IX+4)
0190  ;               "Schleifenzaehler laden"
0200          LD      B,(IX+12)
0210  ;               "plus 1"
0220          INC     B
0230          RRCA
0240          JR      C,TBIT
0250          RRCA
0260          JR      C,SBIT
0270          RRCA
0280          JR      NC,ENDE
0290  ;               "Ab hier RES"
0300  RBIT    XOR     A
0310          CPL
0320  LOOP1   RLA
0330          DJNZ    LOOP1
0340          AND     (IX+20)
0350          JR      ENDE
0360  ;               "Ab hier SET"
0370  SBIT    XOR     A
0380          SCF
0390  LOOP2   RLA
0400          DJNZ    LOOP2
0410          OR      (IX+20)
0420          JR      ENDE
0430  ;               "Ab hier BIT"
0440  TBIT    XOR     A
0450          SCF
0460  LOOP3   RLA
0470          DJNZ    LOOP3
0480          AND     (IX+20)
0490          JR      Z,ENDE
0500          LD      A,1
0510  ENDE    LD      C,A
0520          EI
0530          RET
0540          END
```

Das Maschinenprogramm zur Bitmanipulation ist auf jeder Startadresse lauffähig.

Parameter +1, und das Ergebnis ist dann (B< >255) größer als 255.

Man kann also sehr leicht noch andere Bitmanipulationen einhängen (x AND n, x XOR n und so weiter), die in BASIC teilweise noch wesentlich schwieriger zu programmieren sind, in Maschinensprache aber keinerlei Mühe bereiten. Hierfür kann man noch weitere Bit-Abfragen im Operationsdekoder (vor Zeile 280) vornehmen (für 8,16 ...) und die entsprechenden Mini-Routinen anspringen. So nutzt man die gleiche Technik aus, um die Adressen der Operanden aus der Parameterliste der DEF FN zu 'klauben'. Das Betriebssystem stellt dafür nämlich einen Vektor zur Verfügung DEFADD auf Adresse 23563, der auf die Adresse der Argumente einer gerade ausgeführten anwenderdefinierten Funktion zeigt.

Für XOR könnte das beispielsweise so aussehen:

```
272 JR C,RBIT
274 RRCA
276 JR C,MANI
278 RRCA
280 JR NC,ENDE
MANI : DEC B
       XOR B ;oder OR,
       LD B,0
       JR ENDE
```

ct

RAMs und EPROMs besonders BILLIG!

2732K/250 nS Mitsubishi . . . . . 8,20 DM/St.
2764K/250 nS Hitachi . . . . . 5,90 DM/St.
27128K/250 nS Mitsubishi . . . . . 7,35 DM/St.
27256/250 nS Hitachi . . . . . 12,70 DM/St.
4164-150 nS Toshiba . . . . . 3,00 DM/St.
41256-150 Toshiba . . . . . 7,65 DM/St.
6116LP-3 Hitachi . . . . . 4,20 DM/St.
6264LP-15 Hitachi . . . . . 7,75 DM/St.

TEAC-Floppy-Laufwerke
TEAC 55BV 0,5 MB 2×40 Track . . . . . 320,00 DM
TEAC 55FV 1,0 MB 2×80 Track . . . . . 360,00 DM

WINCHESTER-Laufwerk
20 MB NEC D 5126 incl. DTC-Controller . . 2.120,00 DM

IBM-Interface-Karten
XT8088 Mainboard, bis 640K aufrüstbar . . . . 330,00 DM
384KB Multifunktions-Karte (OK-RAM) . . . . . 280,00 DM
Multi I/O-Karte . . . 270,00 DM
Color-Grafik-Karte . 175,00 DM
Mono-Grafik/Printer-Karte (Herkules) . . . . . 220,00 DM
512K RAM-Karte . . 135,00 DM
Parallel-Karte . . . 60,00 DM
RS-232C-Karte . . . 85,00 DM
Floppy-Disk-Controller für 4 Laufwerke . . . . . 110,00 DM

Preise incl. MWSt.
Bei größeren Abnahmemengen sind wir preisflexibel!

IBM-Kompatibel
- 8088 CPU, 8087 (Option)
- 256-640K auf Mainboard
- 2×360K Mitsubishi Laufwerke
- Multi I/O Karte
  - incl. Controller f. 2 Laufwerke
  - incl. serieller + paralleler Schnittstelle und Gameport
  - Batteriegepufferte Uhr/Kalendar
- Mono/Grafikkarte (Herkules)
- Deutsche DIN-Tastatur
- DOS 3.0
- Aufschlag für 12" Monochrome Monitor, grün . . . . . 320,00 DM

DM 2.150,00

KWEM

Postf. 2528, 34 Göttingen, Tel.: 0551/44077-78, Telex 965202

Die "TURBOS" von ebh-computer gmbh

— bis zu 40 % mehr Leistung durch den Turbo-Betrieb!

— Turbo heißt noch mehr Software schneller verarbeiten können.

Der kompl. ebh-turbo-pc
incl. hochauflösendem Monitor und Tastatur
für . . . . . **2998,— DM**

Der ebh-at
mit 1 MB-Arbeitsspeicher
für . . . . . **7750,— DM**

ebh ist IBM-kompatibel!

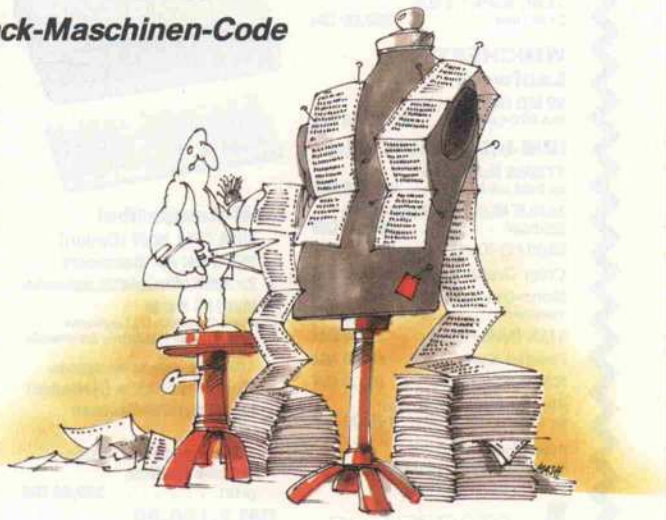
ebh-pc gibt es im guten Fachhandel!

— wegen des Service —

ebh computer gmbh
Wolfsberger Straße 7
4710 LÜDINGHAUSEN
Telefon 02591/4734

Compiler selbstgeschneidert

Teil 4: Erzeugung von Stack-Maschinen-Code



Helmut Richter

Diese als Compiler-Praxis angekündigte Serie befaßt sich jetzt zum ersten Mal mit der Erzeugung von Maschinencode, allerdings noch nicht bis zur letzten Konsequenz. Im ersten Schritt erzeugt der Compiler einen Zwischencode, der nicht von einer real existierenden CPU ausgeführt werden kann. Die Wege von Z80-CP/M-Usern und anderen Compilerbauern brauchen sich noch nicht zu trennen. Der Zwischencode ist für einen nur scheinbar vorhandenen Prozessor gedacht, der in erster Linie mächtige Befehle zur stapelartigen Verwaltung von Daten kennt. Die Codeerzeugung für diese virtuelle Stack-Maschine geschieht praktisch im Parser, der dazu an wohlüberlegten Stellen mit Aufrufen der Prozedur GEN versehen wird.

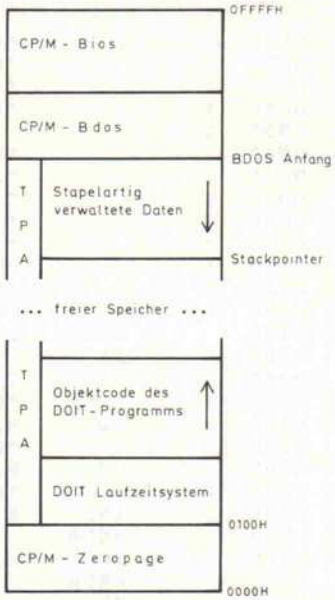
Erstaunlich ist der Umstand, daß auch in dieser Folge lediglich die Eigenheiten von Turbo-Pascal in die Weiterentwicklung des DOIT-Compilers einfließen, Kenntnisse über die Zielmaschine aber, deren CPU und Befehlssatz, nicht nötig sind. Vom Betriebssystem werden nur die Funktionen 'Lese Zeichen von Tastatur' (read console) sowie 'Schreibe Zeichen auf Bildschirm' (write console) benutzt.

Zunächst aber benötigt man eine erste grobe Idee, wie überhaupt Maschinenprogramm und Daten für einen störungsfreien Programmlauf im Hauptspeicher abzulegen sind.

Wie so oft, gibt es auch hier zwei grundsätzliche Fälle, je nachdem, ob die zu übersetzende Sprache rekursive Unterprogramme zuläßt oder nicht. Kennt die Programmiersprache keine Rekursion, wie etwa BASIC oder Fortran, so besteht die Möglichkeit, die Daten statisch in das laufende Maschinenprogramm einzuflechten, am einfachsten da, wo sie im Programm zum ersten Mal erwähnt werden. Vor jedem Datenblock muß dann natürlich ein Jump hinter die Daten stehen, damit die CPU ihren nächsten Befehl findet.

Ist jedoch Rekursion zugelassen, so müssen bei jedem Unterprogrammaufruf neue lokale Daten dynamisch angelegt werden, sofern das Unterprogramm solche hat. Die lokalen Variablen der höheren Rekursionsstufe dürfen weder überschrieben noch darf auf sie zugegriffen werden.

In diesem Fall bietet es sich an, nicht nur die Rücksprungadressen, sondern auch die Daten auf dem CPU-Stack abzulegen. Bei jedem Unterprogrammaufruf reserviert man sich den benötigten Speicherplatz durch eine Dekrementierung des Stackpointers auf dem Stack. Der DOIT-Compiler arbeitet nach diesem Prinzip, ähnlich wie auch Turbo-Pascal oder Modula-2. Diese Art der Datenverwaltung prägt den erzeugten Code so stark, daß sich fast alle Operationen auf den Stack beziehen und ein hypothetischer Prozessor für diesen Code *Stack-Maschine* genannt wird.



Das DOIT-Programm belegt einen festen Bereich innerhalb des Arbeitsspeichers, während die lokalen Daten bei Unterprogrammaufrufen nach unten wachsen. Wachsen sie in den Programmbereich, stürzt das Programm ab.

Was is'n Stack-Maschin?

Ihre Arbeitsweise ist denkbar einfach: Nehmen wir an, das Statement 'A:=B+1;' soll ausgeführt werden. Dann ist folgendes zu tun:

1. Lade Variable B von ihrem Speicherplatz auf den Stack.
2. Lade Konstante 1 auf den Stack.
3. Addiere die beiden obersten Stack-Elemente, entferne sie vom Stack und schreibe das Resultat wieder auf den Stack.
3. Sichere oberstes Stack-Element auf den Speicherplatz von A.

Nr	Mnemonik	Bedeutung
0	LoadInTConst	Lade direkte Integerkonstante auf den Stack
1	LoadCharConst	wie 0 nur mit Character
2	Operation	arithmetische/logische Operation
3	LoadIntVar	Integervariable auf den Stack laden
4	LoadCharVar	wie 3 mit nur Character
5	SaveIntVar	Integervariable vom Stack auf ihren Speicherplatz schreiben
6	SaveCharVar	wie 5 mit Character
7	Call_Proc	Unterprogrammaufruf
8	DECR_SP	Stackpointer (SP) erniedrigen
9	Jump	Unbedingter Sprung (im Zwischencode)
10	Jump_Cond	Bedingter Sprung (IF..THEN..)
11	Call_RTsystem	Laufzeit-Systemaufruf
12	Return	Rückkehr von Unterprogrammruf
13	Save_BP	Alten Basepointer auf Stack sichern
14	Init_SP_BP	Initialisierung der Stackzeiger

Diese 15 Befehle der Stackmaschine genügen, um alle DOIT-Statements ausführen zu können.

Eine Stack-Maschine muß also insgesamt folgendes leisten:

Operationen auf den Daten:
- Laden einer Variablen von ihrem Speicherplatz im Stack auf die Spitze des Stacks,

- Sichern einer Variablen von der Stack-Spitze auf ihren Ablageplatz im Stack.
- Arithmetische und logische Operationen
- Ein- und Ausgabe von Daten Operationen auf dem Stack-Maschinen-Programm:
- Unterprogrammaufrufe
- Returns
- bedingte Sprünge
- unbedingte Sprünge

Nun reicht es gewiß nicht aus, benötigte Variablen beim Betreten eines Unterprogramms einfach auf den Stack zu schreiben, im Handumdrehen wäre die Übersicht und damit auch die Information verloren. Eine genaue Stack-Struktur muß her, die neben den Variablen auch Rücksprungadressen von Unterprogrammaufrufen, Operanden und Operationsergebnisse eindeutig verwaltet.

Dazu eine Überlegung. Zu Beginn eines DOIT-Maschinenprogrammes muß das Programm zuerst den Stackpointer auf die höchste frei verfügbare Speicheradresse setzen. Dies ist bei CP/M die Adresse des BDOS-1. Nun wird der Stackpointer so oft dekrementiert, wie das Hauptprogramm Datenbytes hat.

Da der Quelltext bei der Übersetzung nur einmal durchgelesen wird, ist es zweckmäßig, die Variablen in der Reihenfolge ihrer Deklarationen auf den Stack zu legen. Der Stackpointer zeigt immer auf den nächsten freien Platz des Stacks.

Haben die Daten eine einheitliche Länge, beispielsweise zwei Byte, so kann man ihre Position im Stack einfach durch die Position ihrer Deklaration erkennen:

```
VAR i : INT; (* 1. *)
    j : INT; (* 2. *)
    k : INT; (* 3. *)
```

Variable k, Nummer drei in der Deklarationsfolge, liegt bei einer Einheitslänge von zwei Byte damit auf BDOS-5 und BDOS-6.

Im Compiler ist die Position eines Datums durch die Variable DatenNr vertreten, für ihren richtigen Wert ist die Prozedur MakeData verantwortlich. Wird sie in der Symboltabelle vermerkt, so kann man bei jedem Auftreten einer Variablen sofort ihren Ablageplatz relativ zum Basiszeiger angeben.

Den Basiszeiger auf den Datenbereich des aktivierten (Unter-) Programms nennt man im Compilerbau auch noch Basepointer.

Prozedursegment

Die bisherige Stack-Verwaltung ist noch recht simpel, funktioniert jedoch nur bei Hauptprogrammen. Ungeklärt ist noch, wie man beim Wechsel von einem Unterprogramm zum anderen die zugehörigen lokalen Daten erreicht. Zudem läßt sich bei rekursiven Prozeduren nicht im vorhinein sagen, wie viele Aufrufe stattfinden und wie viele lokale Datensätze zu verwalten sind. Eine dynamische Datenverwaltung muß her.

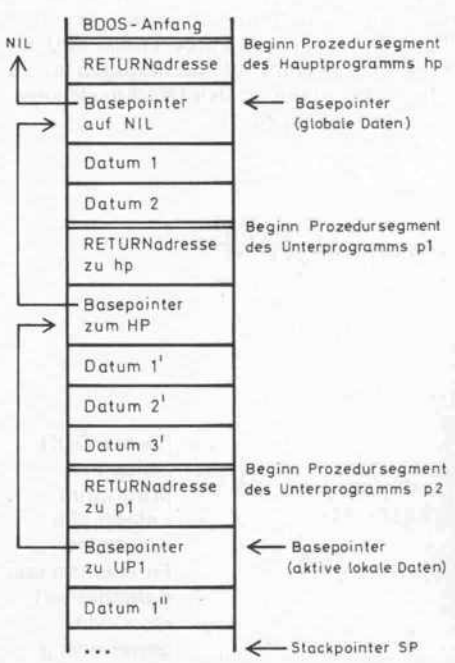
Dazu geht man völlig analog vor, merkt sich jedoch zusätzlich den Basepointer des vorangehenden Unterprogramms und die eigene Rücksprungadresse. Einen so organisierten Datenbereich nennt man Prozedursegment, activation record oder auch Aktivierungs-Record.

In der skizzierten Situation fällt auf, daß jedes Prozedursegment mit einem Zeiger auf den Beginn des nächst höheren Prozedursegmentes beginnt. Der Basepointer zeigt auf diese Speicherstelle. Diese Art der Verzeigerung von Prozedursegmenten wird als dynamischer Link (dynamic link) bezeichnet. Er spiegelt die Reihenfolge der tatsächlichen Unterprogrammaufrufe wider und liefert bei Beendigung des aktiven Unterprogramms den richtigen alten Basepointer.

Diese Konstruktion des Stacks hat jetzt nur noch eine Schwachstelle. Die Sprache DOIT erlaubt den Unterprogrammen, auf die Daten des Hauptprogrammes ohne besondere Ankündigung zuzugreifen. Man muß somit jederzeit das Prozedursegment des Hauptprogrammes auffinden können. Wenn dies dann auch noch schnell geht – um so besser. Im Compiler gibt es daher einen zweiten Basepointer, der ständig auf die globalen Daten gerichtet bleibt.

Das Prozedursegment des Hauptprogramms weist eine Besonderheit auf. Es besitzt ebenfalls eine Rücksprungadresse. Der Grund dafür ist reine Bequemlichkeit. Sowohl Haupt- als auch Unterprogramm können so mit exakt denselben Analyse- und Codegenerierungsroutinen bearbeitet werden.

Dazu ist ein *Super*-Hauptprogramm nötig, das Stack- und Basiszeiger initialisiert, das Hauptprogramm aufruft und nach Ablauf des DOIT-Programms einen CP/M-Warmstart macht. Dann ist eine Unterscheidung zwischen Haupt- und Unterprogrammen nicht mehr nötig.



Das Hauptprogramm und jeder Unterprogrammaufruf hat seinen eigenen Datenbereich auf dem CPU-Stack. Der Basepointer zeigt hier auf die aktiven, lokalen Daten des letzten Unterprogrammaufrufes.

Codegenerierung

Der erste Schritt der Codegenerierung besteht darin, die DOIT-Statements in entsprechende Sequenzen von Stack-Maschinen-Befehlen zu übersetzen. Hierbei kommt dem Compiler der Vorteil von LL(1)-Grammatiken und des Parsings durch rekursiven Abstieg zu Hilfe. Ist ein Übersetzungssachverhalt beim Parsen vollständig erfaßt, so wird an Ort und Stelle die Erzeugung des Objektcodes durch Angabe der benötigten Stack-Maschinen-Operation veranlaßt. Die richtige Reihenfolge der erzeugten Befehle wird dabei durch die rekursive Struktur des Parsers sozusagen von selbst einbehalten.

Am Beispiel 'i:=i+1' wird der Ablauf der Analyse und die Codegenerierung deutlich. Die Prozedur Statement (in PARS-003.INC) ruft gemäß der Ableitungsregel 'statement ::= identifier := expression' Expression auf. Weiter geht es nach den Regeln 'expression ::= term' und 'term ::= factor' mit den Prozeduren Term und Factor. Und Factor findet wegen 'factor ::= identifier' den Identifier i. Jetzt kann schon Code erzeugt werden:

```
GEN(3,0,. . .,'LoadIntVar . . .'); bzw.
GEN(4,0,. . .,'LoadCharVar ..');
```

je nach dem Typ von i.

Das dem Identifier folgende Plus-Zeichen bewirkt den Rücksprung zu Term und von dort zu Expression.

Die nächste Regel lautet 'expression ::= term + term', und der Parser steigt über Term und Factor zum zweiten Mal ab und stößt diesmal in Factor auf eine Integer-Konstante. Der nächste Code wird generiert durch die Aufrufe:

```
GEN(0,0,IWERT,'LoadIntConst . . .');
bzw.
GEN(1,0,IWERT,'LoadCharConst . . .');
```

Das folgende Symbol ist ein Semikolon, und über Term geht es zurück nach Expression. Hier wird die Erzeugung des Codes veranlaßt, der die Addition ausführt:

```
GEN(2,0,2,'Addition . . .'),
```

worauf in Statement der Code für die Sicherung des Ergebnisses auf den Speicherplatz von i erzeugt wird:

```
GEN(5,. . .,DatenAdresse,'Save IntVar'. . .);
```

Es empfiehlt sich nach wie vor, die Compiler-Prozeduren genau nachzuvollziehen. Die exakten Abläufe können in allen Einzel-

heiten nur dem Programm-Listing entnommen werden.

In Stack-Maschinen-Befehlen sieht dies alles dann so aus:

```
LoadIntVar    0 1 ;
LoadIntConst  0 1 ;
Operation     0 2 ;
SaveIntVar    0 1 ;
```

Im Compiler besteht jeder Stack-Maschinen-Befehl aus drei Zahlen x,y und z. Die erste ist die Nummer des Befehls, und die zweite Zahl gibt die Leveldifferenz beim Zugriff auf Variablen an. In einem Unterprogramm bedeutet y=1 den Zugriff auf die Daten des Hauptprogramms und y=0 den Zugriff auf die eigenen lokalen Daten. Die dritte Angabe ist entweder eine direkte oder eine basepointer-relative Adresse oder ein Datum.

Der Stack-Maschinen-Code kann als compiler-interner Zwischencode aufgefaßt werden, der nach außen nicht in Erscheinung tritt. Der Compiler kann ihn aber auf eine Datei schreiben, am Bildschirm zeigen oder sogar ausdrucken, um die Codegenerierung transparent zu machen.

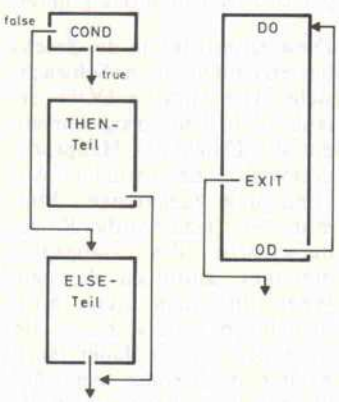
Anders liegen die Dinge zum Beispiel beim UCSD-Pascal-Compiler. Er erzeugt nur Zwischencode, den sogenannten p-Code, der ebenfalls für eine hypothetische Stack-Maschine gedacht ist. Erst zur Laufzeit wird er von einem für die CPU des Rechners geschriebenen Interpreter ausgeführt.

Der Code der Stack-Maschine spielt im DOIT-Compiler also nur die Rolle des Übergabeparameters für die Prozedur GEN. Speichert man ihn zusätzlich in einer Variablen (ZCODE), kann beispielsweise beim Befehl Nr. 3 eine kleine lokale Optimierung durchgeführt werden. War nämlich der vorangehende Befehl ein SaveIntVar auf dieselbe Variable, so befindet sie sich ja noch als letztes Datum auf dem Stack, und es reicht aus, den Stackpointer zu dekrementieren, anstatt einen umständlichen Ladeprozess auszuführen.

Backpatching

Ein interessantes Problem für Compilerbauer ist die Erzeugung von Sprüngen im Zwischencode. Dabei muß zwischen Sprüngen nach vorne und Sprüngen nach hinten unterschieden werden. Vorwärtssprünge werden beim EXIT- und beim IF-THEN-Statement nötig, während Rückwärtssprünge nur im Code von DO..OD-Schleifen auftreten.

Kommt ein IF-Statement zur Übersetzung, so muß abhängig von der Auswertung der Bedingung ein Programmblock durchlaufen und ein anderer übersprungen werden. Zur Ausführung kommen darf ja nur einer der beiden Zweige. Leider sind in diesem Stadium der Übersetzung die Ansprungadressen noch nicht verfügbar. Man weiß ja nicht, wie viele DOIT-Statements Then-Zweig und Else-Zweig umfassen und wie viele Zwischencodebefehle zu ihrer Übersetzung benötigt werden.



Obwohl DOIT kein GOTO kennt, gibt es Vorwärts- und Rückwärtssprünge im übersetzten Code. Ursache sind die bedingte Anweisung, der EXIT-Befehl und die DO..OD-Schleife.

Also merkt der Compiler sich die Adresse des Absprungbefehls in einer Variablen, beispielsweise L0 durch

```
L0:=MakeLabel;
```

und trägt nachträglich beim Auffinden des Ansprungpunktes die momentane Adresse durch

```
FixUp(L0);
```

an der Absprungstelle im übersetzten Code ein. MakeLabel liefert dabei die aktuelle Zwischencodeadresse. Dieses Verfahren, nachträglich Vorwärtssprünge einzutragen, heißt Backpatching.

Um nun eine zu starke Belastung des Laufwerks durch ständiges Hin- und Hersuchen zu vermeiden, speichert der Compiler alle Patch-Informationen in einem Array namens PatchFeld, Typ PatchInfo, und arbeitet dieses Feld am Ende des Parsing-Prozesses auf einmal ab. MakeLabel liefert diesmal keine Adresse, sondern einen Index für ein freies PatchFeld-Record.

Zweimal EXIT

Die EXIT-Sprünge stellen eine weitere Komplikation dar. Sie können sowohl in als auch außerhalb von Schleifen auftreten. Stehen sie in einem DO..OD-Statement, so muß ein Sprung hinter das OD generiert werden, um die Schleife zu verlassen. Kommen sie außerhalb vor, wirken sie wie ein Return, das heißt, es muß ein Sprung auf das (Unter-)Programmende generiert werden. Zur Lösung dieses Problems wendet der Compiler ein besonderes Verfahren an.

Die Verwaltung der EXITs erfolgt mit Hilfe einer linearen Liste, die exakt so organisiert ist wie die Symboltabelle. Trifft der Parser auf ein DO-Symbol, so legt er mit Hilfe von InitExitListe eine lokale Tabelle für eventuelle EXIT-Sprünge an, genau so, wie Block eine neue Symboltabelle für die lokalen Variablen erzeugt.

Alle vorgefundenen EXITs werden nun in diese Liste eingetragen und am Schleifenende mit FixUpExitListe in PatchFeld überführt. Die lokale Liste wird dann verworfen. Ist am (Unter-)Programmende der Zeiger Exit-Sprung ungleich nil, so hat ein EXIT die Wirkung eines Returns. Wieder trägt FixUpExit-Liste die Patch-Informationen ins PatchFeld ein.

Rückwärtssprünge treten nur beim DO...OD-Statement auf und stellen den einfacheren Fall dar; zur Compile-Zeit steht die Ansprungadresse ja schon zur Verfügung. Im Zwischencode geschieht das Backpatching folgendermaßen:

Zu Beginn der Schleife reserviert man sich einen Index auf PatchFeld mittels MakeLabel, speichert ihn in der lokalen Variablen L0 und bearbeitet die Schleifen-Statements. Nach Vorfinden des OD muß ein Sprung auf den Schleifenanfang generiert werden. PatchFeld[L0].Z_Adresse enthält nicht wie beim Vorwärtssprung die Patchadresse, sondern den Patchwert, das Sprungziel und muß auf Z_Patch umkopiert werden. Die Patchadresse ist nun die aktuelle ZCode_Adresse.

Unterstützung

Sämtliche Arithmetik-, Logik- und Systemoperationen müssen bei der Simulation der Stack-Maschine vom 'Stack-Maschinen-Prozessor' geliefert werden. Da normale CPUs solche komplexen Operationen nicht als Maschinenbefehle zur Verfügung stellen, müssen sie durch Unterprogramme realisiert werden.

Hier gibt es zwei Möglichkeiten, die notwendigen Unterprogramme mit dem übersetzten Programm zu verbinden:
1. Wie in Turbo-Pascal wird das komplette Laufzeitsystem als Block an den Objektcode angekoppelt.

```
MODULE Test;
(* Allgemeines Testmodul für DOIT *)
VAR x       : INT;
    i       : INT;

PROC ReadInt;
BEGIN
  WRITE '?'; READ x;
END (* ReadInt *);

BEGIN (* Hauptprogramm *)
  DO ReadInt;
     IF x < 0 THEN EXIT; FI;
     x := 6 / x;
     WRITE x;
  OD;
END.
```

Dieses DOIT-Beispielprogramm enthält alle wichtigen Funktionen und demonstriert die Codegenerierung.

2. Nur die im übersetzten Programm tatsächlich benötigten Funktionen werden an das Objektmodul angebunden (gelinkt). Dies geschieht zum Beispiel bei Modula-2 und MT+Pascal.

Obwohl der zweite Weg der perfektere ist und beide für die Sprache DOIT recht einfach realisiert werden können, verwendet DOIT die Methode des kopierten Laufzeitblocks. Der Grund ist offensichtlich:

Dieses Konzept ist einfacher zu implementieren. Wird Objektcode bei einer Übersetzung gewünscht, kopiert der Compiler einfach das Laufzeitsystem (Name: DCRTS.LIB) zu der COM-Datei dazu.

Die Adressen der verschiedenen Unterprogramme aus der Lib sind im Compiler als Konstanten vereinbart und werden bei der Kompilation einfach in den Objektcode eingesetzt. Zudem beginnt das Laufzeitsystem wie bei CP/M auch mit einer Vektorenleiste.

Der nächste und letzte Teil dieser Serie wird sich dem endgültigen Einstieg in die Praxis, nämlich dem Laufzeitsystem und der Erzeugung von Assembler- und Maschinencode widmen.

```
Adresse OpCode              Level  Adresse/Wert
-------------------------------------------------

    0 : Init_SP_BP            0      57349 ; Stack/Basepointer initialisieren
    1 : Call_Proc             0          3 ; Aufruf des Hauptprogramms
    2 : Jump                  0          0 ; Warmstart
    3 : Save_BP               0          0 ; Basepointer sichern, neuen laden
    4 : Decr_SP               0          2 ; Platz fuer Daten schaffen
    5 : Decr_SP               0          2 ; Platz fuer Daten schaffen
    6 : Jump                  0         13 ; Mgl. Unterprogramme ueberspringen
    7 : Save_BP               0          0 ; Basepointer sichern, neuen laden
    8 : LoadCharConst         0         63 ; direkte Characterkonstante
    9 : Call_RTsystem         0         13 ; Character schreiben
   10 : Call_RTsystem         0          4 ; Integer einlesen
   11 : SaveIntVar            1          1 ; und sichern
   12 : Return                1          0 ; Return vom Unterprogrammaufruf
   13 : Call_Proc             1      26653 ; Unterprogramm rufen
   14 : LoadIntVar            0          1 ; Lade Integervariable X
   15 : LoadInTConst          0          0 ; direkte Integerkonstante
   16 : Operation             0         46 ; Kleiner testen
   17 : Jump_Cond             0         19 ; Sprung auf ELSE-Teil
   18 : Jump                  0         26 ; EXIT-Sprung
   19 : LoadInTConst          0          6 ; direkte Integerkonstante
   20 : LoadIntVar            0          1 ; Lade Integervariable X
   21 : Operation             0         28 ; Division durchfuehren
   22 : SaveIntVar            0          1 ; Save Integer X
   23 : LoadIntVar            0          1 ; Lade Integervariable X
   24 : Call_RTsystem         0         10 ; Integer schreiben
   25 : Jump                  0         13 ; DO..OD: Ruecksprung zum DO
   26 : Return                0          0 ; Return vom Hauptprogrammaufruf
```

Das Beispielprogramm in Stack-Maschinen-Befehlen.

```
PROGRAM DoitCompiler;
(* ------------------------------------------------------------ *)
(* Doit-Yourself Compiler                                       *)
(* Autor     : Helmut Richter                                   *)
(* Version   : 2.1                        Datum: 4. Februar 1986 *)
(*  Die  Parser-Prozeduren werden hier nur so  weit noch        *)
(*  einmal gezeigt, wie gegenueber der Fassung in Teil 3        *)
(*  fuer die Codegenerierung Ergaenzungen notwendig sind.       *)
(* ------------------------------------------------------------ *)
(*$U-,C+,A-*)

CONST (*$I SCCONST.INC *)
(*    Adressen der Sprungvektoren aller Funktionen des Run-    *)
(*    Time- Systems (Library)                                  *)
      ReadInt    =  4; (* Systemfunktionen (READ/WRITE)       *)
      ReadChar   =  7; WriteInt  = 10; WriteChar = 13;
      Addop      = 16; (* Arithmetische Operationen           *)
      SUBop      = 19; NEGop     = 22; Multop    = 25;
      DIVop      = 28; MODop     = 31;
      ODDop      = 34; (* Logische Operationen                *)
      EQLop      = 37; NEQop     = 40; GTRop     = 43;
      LSSop      = 46; GEQop     = 49; LEQop     = 52;
(*    Anzahl der "Maschinen"operationen, der virtuellen CPU    *)
      MaxOp      = 14; (* Index beginnt bei 0 !               *)
      Memo : ARRAY[0..MaxOp] OF STRING[13] =
(*         Stackmaschinenbefehl  OpCode                        *)
            ('LoadInTConst ', (*  0 *) 'LoadCharConst', (*  1 *)
             'Operation    ', (*  2 *) 'LoadIntVar   ', (*  3 *)
             'LoadCharVar  ', (*  4 *) 'SaveIntVar   ', (*  5 *)
             'SaveCharVar  ', (*  6 *) 'Call_Proc    ', (*  7 *)
             'Decr_SP      ', (*  8 *) 'Jump         ', (*  9 *)
             'Jump_Cond    ', (* 10 *) 'Call_RTsystem', (* 11 *)
             'Return       ', (* 12 *) 'Save_BP      ', (* 13 *)
             'Init_SP_BP   ');(* 14 *)
      TPA_Anfang     = $100; (* Anfang von CP/M-Programmen:    *)
```

```
(*    Startadresse des Maschinenprogramms: TPA + Laenge RTS   *)
      MCode_Offset   = $200;
      ZCode_Offset   =    0; (* Startadresse des Zwischencode *)
      ASMCode_Offset = $200; (*   " des Assemblercodes        *)
(*    Laenge des Patchfeldes fuer die Rueckwaertsspruenge:    *)
      MaxPatch       =   40;
      Test           = FALSE;

TYPE  (*$I SCTYPE.INC *)
      Objekt_Typ   = (Integer_Objekt, Character_Objekt,
                      Fehler_Objekt); (* verfuegtbare Typen   *)
      ... STR10, Objektklasse u. ObjPtr wie gehabt ...
      Objekt       = RECORD ...
                       CASE kind  : ObjektKlasse OF
                         ...
                         Prozedur : (ZWC_StartAdresse,
                                     ASM_StartAdresse,
                                     COM_StartAdresse: INTEGER);
                         CharVar, IntVar : (DatenAdresse, Vlevel
                                            : INTEGER);
                         Header   : (Last, Down : ObjPtr);
                     END;
(*    Typen fuer die EXIT-Sprung-Liste                         *)
      ExitListenTyp = (Hdr, Blatt);
      ExitPtr      = ^ExitListe;
      ExitListe    = RECORD
                       L    : INTEGER; next : ExitPtr;
                       CASE ElementTyp : ExitListenTyp OF
                         Blatt : ();
                         Hdr   : (Last, Down : ExitPtr);
                     END;
      Str40        = STRING[ 40]; Str7        = STRING[  7];
      Str4         = STRING[  4]; Str2        = STRING[  2];
      Dateitypen   = (SRC,COM,PRN,ZWC,ASM);
(*    Typen fuer Assembler- und Zwischencode                   *)
      ZCode_Instruction = RECORD       (* Zwischencode-Befehl  *)
                       nr : INTEGER;   (* Nummer des Befehls   *)
                       bc : 0..MaxOp;  (* Befehlscode          *)
                       lv : 0..15;     (* Level des Programms  *)
                       ad : Integer;   (* Adresse des Datums   *)
                       km : str40;     (* Kommentar            *)
                     END;
      ASM_Instruction = RECORD         (* Assembler-Befehl     *)
                       nr : Integer;   (* Nummer des Befehls   *)
                       bc : Str4;      (* Operationscode       *)
```

```
                    op1,            (* Operand 1              *)
                    op2: Str7;      (* Operand 2              *)
                    km : Str40;     (* Kommentar              *)
                  END;
    PatchInfo   = RECORD (* Typ fuer das Patchfeld            *)
(*                  Zwischencode-Adresse und -Patchwert       *)
                    Z_Adresse, Z_Patch,
(*                  Assemblercode-Adresse und -Patchwert      *)
                    A_Adresse, A_Patch : INTEGER;
(*                  Indikator fuer Art des Sprunges           *)
                    JPC                : BOOLEAN;
(*                  Maschinencode-Adresse und -Patchwert      *)
                    M_Adresse, M_Patch : INTEGER;
                  END;

VAR   (*$I SCVAR.INC *)
      TopScope, Bottom : ObjPtr;
      ProgrammName     : T_Name;
(*    Variable zur Feststellung der Rekursionstiefe           *)
      level            : Integer;
(*    Variable zur Pruefung der Typengleichheit               *)
      Expression_Typ   : Objekt_Typ;
(*    Variable fuer Zwischen- und Assemblercodebefehle        *)
      ZCode            : ZCode_Instruction;
      ASMCode          : ASM_Instruction;
      ZCode_Adresse,    (* Nachster freier Zwischencodebefehl *)
      MCode_Adresse,    (*  "          freier Maschinenbefehl *)
      ASMCode_Adresse  : INTEGER; (*  "    "   Assemblerbefehl *)
      ExitSprung       : ExitPtr;
(*    Deklarationen der Compilerdateien                       *)
      Source, COM_File, List : TEXT;
      ZWC_File         : FILE OF ZCode_Instruction;
      ASM_File         : FILE OF ASM_Instruction;
(*    Feld zur Markierung geoeffneter Dateien:                *)
      Ist_offen        : ARRAY[DateiTypen] OF BOOLEAN;
(*    Schalter fuer die Ausgabedatei-Generierung:             *)
      COM_Gen, liston, ASM_Gen, ZWC_Gen,
(*    Variable zur Markierung des Bedingten Sprunges          *)
      Jump_on_Condition: BOOLEAN;
(*    Deklarationen fuer die Sprungadressenverwaltung         *)
(*    Patchfeld und sein Zaehler                              *)
      PatchFeld        : ARRAY[1..MaxPatch] OF PatchInfo;
      ipatch,
(*    Fortlaufende Nummer einer Variablen in ihrer Symbol-    *)
(*    tabelle (dient zur Berechnung ihres Stackoffsets)       *)
      DatenNr          : INTEGER;

PROCEDURE Wait;
BEGIN REPEAT UNTIL KeyPressed; END (* Wait *);

FUNCTION AdressZahl(i:integer):REAL;
BEGIN
  IF i < 0 THEN AdressZahl:=65536.0+i ELSE AdressZahl:=0.0+i;
END (* AdressZahl *);

FUNCTION Exist(DateiName:Str40):BOOLEAN;
VAR Datei : File;
BEGIN (* Testet, ob die Datei mit Namen (DateiName) existiert *)
  Exist := FALSE; ASSIGN(Datei,DateiName);
  (*$I-*) RESET(Datei); (*$I+*)
  IF (IOResult = 0) THEN BEGIN CLOSE(Datei); Exist := TRUE; END
  ELSE BEGIN HighVideo;
    WRITELN('Datei ',DateiName,' existiert nicht.'); NormVideo;
  END;
END (* Exist *);

(*$I PARS-001.INC *) (* Alle Scannerprozeduren.              *)
(*$I DC-002.INC   *) (* Codegenerator GEN                   *)
(*$I DC-003.INC   *) (* Hilfsprozeduren zur Codegenerierung *)
(*$I PARS-002.INC *) (* Hilfsprozeduren des Parsers / EXITver.*)
(*$I PARS-003.INC *) (* Prozeduren Expression, Statement.   *)
(*$I PARS-004.INC *) (* Prozeduren Block, Parse.            *)
(*$I DC-007.INC   *) (* Hilfsprozeduren des Compilers.      *)
(*$I DC-008.INC   *) (* Dialogprozeduren des Compilers      *)

BEGIN InitDC;          (* -- DOIT - Hauptprogramm ---------- *)
  REPEAT DisplayMenue; (* -- Hauptschleife des Compilers ---- *)
    IF ch = 'C' THEN Compile;
  UNTIL (ch = 'D') OR (ch = 'X');
  Schliessedateien;
END.

(* ------------------------------------------------------------ *)
(* Modul DC-001.INC   (PARS-001.INC)                  Scanner *)
(* ------------------------------------------------------------ *)
Der Scanner bleibt im wesentlichen unveraendert.
```

```
In InitScanner muessen die Dateimanipulationen entfernt werden.
(Geschehen nun in InitParser). Sie sieht jetzt so aus:

PROCEDURE InitScanner;
BEGIN (* InitScanner *)
  ClrScr; Ctrl := ' '; sym := Null; intval := 0; Charval := ' ';
  errcount := 0; lesen := TRUE; Noerr  := TRUE; ch := ' ';
  SwT(. 1.).s := 'BEGIN';  SwT(. 1.).NR := 32;
  ... Initialisierung der Schluesselworttabelle ...
  SwT(.18.).s := 'WRITE';  SwT(.18.).NR := 41;
  ClrScr; HighVideo; WRITELN('Compiling...');
  WRITELN; NormVideo; Satz := ''; Fuellepuffer;
END (* InitScanner *);

PARAMCOUNT und PARAMSTR muessen vor InitScanner gezogen
werden, um global erreichbar zu sein. (Nur TURBO 2.0)
SYMNAME, PROMPT und alle ihre Referenzen koennen entfernt werden.

(* ------------------------------------------------------------ *)
(* Modul DC-002.INC                              Codegenerator *)
(* ------------------------------------------------------------ *)
PROCEDURE Gen(x, y, z:Integer; Kommentar:str40);
(* Generierung von Stackmaschinen-Code (Rest naechstes Heft)  *)
BEGIN
  IF Noerr THEN BEGIN
    WITH ZCode DO BEGIN
      nr:=ZCode_Adresse; bc:=x; lv:=y; ad:=z; km:=Kommentar;
    END;
    IF ZWC_Gen THEN WRITE(ZWC_File,ZCode);
    ZCode_Adresse := ZCode_Adresse + 1;
  END;
END (* Gen *);

(* ------------------------------------------------------------ *)
(* Modul DC-003.INC       Hilfsprozeduren des Codegenerators *)
(* ------------------------------------------------------------ *)
PROCEDURE FixUp(adresse : INTEGER);
BEGIN (* Backpatching der Spruenge und Unterprogrammaufrufe - *)
  WITH PatchFeld[adresse] DO BEGIN Z_Patch := ZCode_Adresse;
    A_Patch := ASMCode_Adresse; M_Patch := MCode_Adresse;
  END;
END (* FixUp *);

PROCEDURE BackPatch;   (* -- Vorwaertsspruenge patchen ------ *)
  PROCEDURE PatchZWCFile;
  (* Patches per RANDOM-ACCESS in die richtigen Saetze        *)
  (* schreiben. Zugriff erfolgt ueber die Satznummer.         *)
  VAR ZCode2 : ZCode_Instruction;
      i, j   : INTEGER;
  BEGIN
    WRITELN; WRITELN('Zwischencode-File...');
    RESET(ZWC_File); i := 1;
    WHILE i < ipatch DO BEGIN
      WITH ZCode2 DO WITH PatchFeld[i] DO BEGIN
        j := Z_Adresse - ZCode_Offset;
(*      Random Access: gewuenschten Satz suchen und in       *)
(*      ZCode2 lesen. Achtung: der Satzzeiger rueckt einen   *)
(*      Satz vor!                                            *)
        SEEK(ZWC_File, j); READ(ZWC_File, ZCode2);
(*      Patch eintragen, ...                                 *)
        ad := Z_Patch;
(*      ... Satzzeiger nochmal auf alten Sektor positionieren *)
(*      und zurueckschreiben.                                *)
        SEEK (ZWC_File, j); WRITE(ZWC_File, ZCode2);
(*      Patchinfo auf Bildschirm schreiben.                  *)
        WRITELN('Patch Nr: ',i:2,'  Adresse:',Z_Adresse:5,
                '  Patch: ',Z_Patch:5,'     SatzNr: ',j:3);
        i := i + 1;
      END;
    END;
  END (* PatchZWCFile *);

PROCEDURE PatchASMFile;
BEGIN END;
PROCEDURE PatchCOMFile;
BEGIN END;

BEGIN (* BackPatch *)
  WRITELN; WRITELN; WRITE('Backpatching: ');
  IF ZWC_Gen THEN PatchZWCFile; IF ASM_Gen THEN PatchASMFile;
  IF COM_Gen THEN PatchCOMFile;
  WRITELN('-- Fertig mit Backpatching --');
END (* BackPatch *);

PROCEDURE MakeData(VAR adr:INTEGER; Laenge:Integer);
(* Erzeugt Adresse fuer Daten auf Stack *)
```

```
BEGIN (* hier: 2 Byte pro Variable!!! *)
  adr := DatenNr; DatenNr := DatenNr + Laenge;
(* StackPointer (Laenge)-mal decrementieren ! *)
  Gen(8,0,2,'Platz fuer Daten schaffen');
END (* MakeData *);

FUNCTION MakeLabel : Integer;
BEGIN
  WITH PatchFeld[ipatch] DO BEGIN
    Z_Adresse := ZCode_Adresse;
    IF Jump_on_Condition THEN BEGIN
      A_Adresse := ASMCode_Adresse + 3;
      M_Adresse := MCode_Adresse + 4;  JPC := TRUE;
    END ELSE BEGIN
      A_Adresse := ASMCode_Adresse;
      M_Adresse := MCode_Adresse + 1;  JPC := FALSE;
    END;
    Z_Patch   := 0; A_Patch   := 0; M_Patch   := 0;
  END;
  MakeLabel := ipatch; ipatch    := ipatch + 1;
  IF ipatch ) MaxPatch THEN BEGIN HighVideo;
    WRITELN('Patchfeld overflow - mehr als ',MaxPatch:2,
    ' Spruenge - Nothalt - press key');NormVideo; Wait;
  END;
END (* MakeLabel *);

(* ------------------------------------------------------------ *)
(* Modul DC-004.INC (PARS-002.INC) Hilfsprozeduren des Parser *)
(* ------------------------------------------------------------ *)
FUNCTION TypenGleich(Typ1,Typ2:Objekt_Typ):BOOLEAN;
(* Testet auf Typengleichheit. Ist einer der beiden der       *)
BEGIN (* Fehler_Typ, so muss FALSE geliefert werden.          *)
  IF (Typ1 = Fehler_Objekt) OR (Typ2 = Fehler_Objekt) THEN
  TypenGleich := FALSE ELSE TypenGleich := (Typ1 = Typ2);
END (* TypenGleich *);

PROCEDURE ProcCall(Obj:ObjPtr);
(* Bearbeiting eines Unterprogrammaufrufs                     *)
BEGIN (* Ausnahme: keine Adresse sondern Zeiger auf Symbol-   *)
(* tabelle (als Integer getarnt) an GEN uebergeben.          *)
  Gen(7,1,ORD(Obj),'Unterprogramm rufen'); GetSym;
END (* ProcCall *);

FUNCTION  NewObj(k : ObjektKlasse) : ObjPtr;
  ...
(* Neues Element anlegen *)
  ...
    CASE Kind OF
      Nix          : Typ := Fehler_Objekt;
      CharacterCon : BEGIN CWert := CharVal;
                     Typ   := Character_Objekt; END;
      IntegerCon   : BEGIN IWert := IntVal;
                     Typ   := Integer_Objekt; END;
      Prozedur     : BEGIN ZWC_StartAdresse := 0;
                     ASM_StartAdresse := 0;
                     COM_StartAdresse := 0; END;
      CharVar      : BEGIN MakeData(DatenAdresse, 1);
                     Typ := Character_Objekt; END;
      IntVar       : BEGIN MakeData(DatenAdresse, 2);
                     Typ := Integer_Objekt; END;
    END (* CASE Kind OF *);
  ...

FIND, TESTSYM und TESTSEMICOLON bleiben unveraendert. Neu sind:

PROCEDURE NewExitListe;
(* Initialisierung der Liste der EXIT-Spruenge mit Header *)
VAR ExitListenKopf : ExitPtr;
BEGIN   NEW(ExitListenKopf);
  WITH ExitListenKopf^ DO BEGIN
    ElementTyp := Hdr; L := -Maxint; next := NIL;
    Last := ExitListenKopf; Down := ExitSprung;
  END;
  ExitSprung := ExitListenKopf;
END (* NewExitListe *);

PROCEDURE EnterExitListe(Labl : INTEGER);
(* Eintrag eines EXIT-Statements    *)
VAR Eintrag : ExitPtr;
BEGIN   NEW(Eintrag);
  WITH Eintrag^ DO BEGIN
    ElementTyp := Blatt; L := Labl; next := NIL;
  END;
  WITH ExitSprung^ DO BEGIN
    Last^.next := Eintrag; Last := Eintrag;
  END;
END (* EnterExitListe *);
```

```
PROCEDURE FixUpExitListe;
(* "Backpatching" der Exit-Spruenge eines DO..OD-Levels *)
VAR Zeiger : ExitPtr;
BEGIN    Zeiger := ExitSprung^.next;
  WHILE Zeiger () NIL DO BEGIN
    FixUp(Zeiger^.L); Zeiger := Zeiger^.next;
  END;
  ExitSprung := ExitSprung^.Down;
END (* FixUpExitListe *);

(* ------------------------------------------------------------ *)
(* Modul DC-005.INC (PARS-003.INC) Analyseprozeduren          *)
(* ------------------------------------------------------------ *)
PROCEDURE Expression;
(* - Die Pruefung der Typengleichheit in Ausdruecken geschieht*)
(*   mit Hilfe von TYPENGELEICH, der lokalen Variablen        *)
(*   FAKTOR_TYP, TERM_TYP, EXPRESSION_TYP (global).           *)
VAR Operation : Symbol; Faktor_Typ, Term_Typ : Objekt_Typ;

  PROCEDURE factor;
  VAR Obj: ObjPtr;
  BEGIN  TestSym(Lparent,95);
    IF sym = identifier THEN BEGIN Obj := Find(idname);
      WITH Obj^ DO CASE kind OF
        IntegerCon   : Gen(0,0,IWert,
                           'Lade Integerkonstante ' + idname);
        CharacterCon : Gen(1,0,ORD(CWert),
                           'Lade Characterkonstante ' + idname);
        IntVar       : Gen(3,Level-Vlevel,Datenadresse,
                           'Lade Integervariable ' + idname);
        CharVar      : Gen(4,Level-Vlevel,Datenadresse,
                           'Lade Charactervariable ' + idname);
        Prozedur     : error(21,
                             'Prozeduren hier nicht erlaubt');
        ELSE         ;
      END;
      Faktor_Typ := Obj^.Typ; GetSym;
    END ELSE IF sym = Intcon THEN BEGIN
      Faktor_Typ := Integer_Objekt;
      Gen(0,0,IntVal,'direkte Integerkonstante'); GetSym;
    END ELSE IF sym = Charcon THEN BEGIN
      Faktor_Typ := Character_Objekt;
      Gen(1,0,ORD(CharVal),'direkte Characterkonstante'); GetSym;
    END ELSE IF sym = Hexcon  THEN BEGIN
      Faktor_Typ := Character_Objekt;
      Gen(1,0,ORD(CharVal),'direkte Hexkonstante'); GetSym;
    END ELSE IF sym = lparent THEN BEGIN GetSym;
      Expression; Faktor_Typ := Expression_Typ;
      IF sym = Rparent THEN GetSym ELSE error(7,'")" erwartet');
    END ELSE BEGIN
      Faktor_Typ := Fehler_Objekt; Error(18,'Ausdruck erwartet');
    END;
  END (* Factor *);

  PROCEDURE Term;
  VAR mulop : Symbol; Faktor1_Typ : Objekt_Typ;
  BEGIN    Factor; Term_Typ := Faktor_Typ;
    WHILE (times (= sym) AND (sym (= modsy) DO BEGIN
      mulop := sym; GetSym; Factor;
      IF TypenGleich(Term_Typ,Faktor_Typ) THEN BEGIN
        Term_Typ := Faktor_Typ;
        CASE mulop OF
          times : Gen(2,0,MULTop,'Multiplikation durchfuehren');
          divsy : Gen(2,0,DIVop,'Division durchfuehren');
          modsy : Gen(2,0,MODop,'Modulo durchfuehren');
        END (* CASE mulop OF ... *);
      END ELSE BEGIN Term_Typ := Fehler_Objekt;
        Error(50,'in Term');
      END (* IF TypenGleich THEN ... *);
    END (* WHILE (times (= ... *);
  END (* Term *);

BEGIN (* Expression *)
  IF (plus (= sym) AND (sym (= minus) THEN BEGIN
    Operation := sym; GetSym; Term;
    IF Operation = minus THEN
      Gen(2,0,NEGop,'Vorzeichen Minus behandeln');
  END ELSE Term;
  Expression_Typ := Term_Typ;
  WHILE (plus (= sym) AND (sym (= minus) DO BEGIN
    Operation := sym; GetSym; Term;
    IF TypenGleich(Expression_Typ,Term_Typ) THEN BEGIN
      Expression_Typ := Term_Typ;
      IF Operation = plus THEN
        Gen(2,0,ADDop,'Addition durchfuehren')
      ELSE Gen(2,0,SUBop,'Subtraktion durchfuehren');
    END ELSE BEGIN
```

```
      Expression_Typ := Fehler_Objekt; Error(50,'in Expression');
    END (* IF TypenGleich THEN ... *);
  END (* WHILE (plus ,= ... *);
END (* Expression *);

PROCEDURE Condition;
VAR relop          : Symbol; Condition_Typ : Objekt_Typ;
BEGIN
  IF sym = odd THEN BEGIN Getsym; Expression;
    IF Expression_Typ () Integer_Objekt THEN
      Error(51,'Nach ODD Integer erwartet');
    Gen(2,0,ODDop,'ODD-Test durchfuehren');
  END ELSE BEGIN
    Expression; Condition_Typ := Expression_Typ;
    IF (eql (= sym) AND (sym (= geq) THEN BEGIN
      relop := sym; Getsym; Expression;
      IF NOT TypenGleich(Condition_Typ,Expression_Typ) THEN
        Error(50,'Typenungleichheit in Condition');
      CASE relop OF
        eql : Gen(2,0,EQLop,'Gleichheit testen');
        neq : Gen(2,0,NEQop,'Ungleichheit testen');
        lss : Gen(2,0,LSSop,'Kleiner testen');
        geq : Gen(2,0,GEQop,'Groesser-gleich testen');
        gtr : Gen(2,0,GTRop,'Groesser testen');
        leq : Gen(2,0,LEQop,'Kleiner-gleich testen');
      END (* Case *);
    END ELSE error(20,'Relation erwartet');
  END (* IF sym = odd THEN ... *);
END (* Condition *);

PROCEDURE Statement;
VAR Obj : ObjPtr; L0, L1 : INTEGER; Folgesymbole : SET OF symbol;

  PROCEDURE IFstatement;
  LABEL 99;
  VAR L0, L1 : INTEGER; NurThenTeil : BOOLEAN;
  BEGIN   NurThenTeil := FALSE;
    GetSym; Condition;
    IF sym = THENsy THEN BEGIN NurThenTeil := TRUE; GetSym;
  (*  Konditionaler Sprung hinter den THEN-Block, falls   *)
  (*  CONDITION = FALSE.                                  *)
      Jump_on_Condition := TRUE;
      L0 := MakeLabel; Gen(10,0,0,'Sprung auf ELSE-Teil');
      Jump_on_Condition := FALSE;
      Folgesymbole := (.ELSEsy, FIsy, ENDsy, period, eofsy.);
      REPEAT   Statement;   UNTIL sym in Folgesymbole;
      IF sym = ELSEsy THEN BEGIN
  (*    Sprung hinter den ELSE-Block nach durchlaufen     *)
  (*    des THEN-Zweiges.                                 *)
        L1 := MakeLabel;
        Gen(9,0,0,'Sprung hinter ELSE-Teil');
  (*    Ansprungadresse des Sprungs ueber den THEN-Block  *)
  (*    eintragen (entspricht CONDITION = FALSE)          *)
        FixUp(L0); NurThenTeil := FALSE; Getsym;
        Folgesymbole := (.FIsy, ENDsy, period, eofsy.);
        REPEAT Statement;
          IF sym = ELSEsy THEN BEGIN
            Error(16,'FI erwartet'); GOTO 99; END;
        UNTIL sym in Folgesymbole;
  (*    Ansprungadresse des Sprungs ueber den ELSE-Block  *)
  (*    eintragen.                                        *)
        FixUp(L1);
      END;
      IF NurThenTeil THEN FixUp(L0);
      IF sym = FIsy THEN GetSym ELSE error(16,'FI erwartet');
    END ELSE error(15,'THEN erwartet');
  99:
  END (* IFstatment *);

BEGIN (* Statement *)   TestSym(semicolon,98);
  IF sym = Identifier THEN BEGIN Obj := Find(idname);
    IF Obj^.kind = Prozedur THEN ProcCall(Obj)
    ELSE IF Obj^.kind = IntVar THEN BEGIN GetSym;
      IF sym = becomes THEN GetSym
                       ELSE Error(11,':= erwartet');
      Expression;
      IF NOT TypenGleich(Integer_Objekt,Expression_Typ) THEN
        Error(50,'in Integerausdruck')
      ELSE WITH Obj^ DO Gen(5,Level-vlevel,Datenadresse,
                            'Save Integer ' + Name);
    END ELSE IF Obj^.kind = CharVar THEN BEGIN GetSym;
      IF sym = becomes THEN GetSym
                       ELSE Error(11,':= erwartet');
      Expression;
      IF NOT TypenGleich(Character_Objekt,Expression_Typ)
      THEN Error(50,'in Characterausdruck')
      ELSE WITH Obj^ DO Gen(6,Level-vlevel,Datenadresse,
                            'Save Character ' + Name);
    END ELSE IF Obj^.Kind = NIX THEN BEGIN GetSym;
      IF sym = Semicolon THEN ProcCall(Obj)
      ELSE BEGIN (* Variablenname / Zuweisung *)
        IF sym = becomes THEN GetSym
                         ELSE Error(11,':= erwartet');
        Expression;
      END;
    END;
  END ELSE IF sym = IFsy THEN IFstatement
  ELSE IF sym = DOsy THEN BEGIN
    Folgesymbole := (.ODsy, ENDsy, period, eofsy.);
(*  Ansprungadresse fuer das Ende der DO ... OD-Schleife  *)
    L0 := MakeLabel;
(*  Liste fuer moegliche Exit-Spruenge anlegen            *)
    NewExitListe; GetSym;
    REPEAT   Statement;   UNTIL sym in Folgesymbole;
    IF sym = ODsy THEN GetSym ELSE error(17,'OD erwartet');
(*  Sprung rueckwaerts zum Anfang des DO .... OD-Blocks   *)
(*  (Adresse vom DO im Patch-Feld umsetzen)               *)
    WITH PatchFeld[L0] DO BEGIN
      Z_Patch   := Z_Adresse; Z_Adresse := ZCode_Adresse;
      A_Patch   := A_Adresse; A_Adresse := ASMCode_Adresse;
      M_Patch   := M Adresse-1; M_Adresse := MCode_Adresse+1;
    END;
    Gen(9,0,0,'DO..OD: Ruecksprung zum DO');
(*  Falls zwischen DO und OD EXIT(s) vorlag(en), muss     *)
(*  die ExitListe abgearbeitet werden.                    *)
    FixUpExitListe;
  END ELSE IF sym = EXITsy THEN BEGIN
(*  EXIT kann auch ausserhalb von DO..OD-Schleifen ver-   *)
(*  wendet werden. Es simuliert dann ein RETURN.          *)
(*  ExitSprung = NIL bedeutet, dass kein DO..OD vorliegt. *)
    IF ExitSprung = NIL THEN NewExitListe;
    EnterExitListe(MakeLabel); Gen(9,0,0,'EXIT-Sprung'); GetSym;
  END ELSE IF sym = READsy THEN BEGIN GetSym;
    IF sym = Identifier THEN BEGIN Obj := Find(idname);
      WITH Obj^ DO BEGIN
        IF Kind = IntVar THEN BEGIN    (* Integervariable *)
(*        UP READ_Integer aufrufen  *)
          Gen(11,0,ReadInt,'Integer einlesen');
(*        Integer-Variable sichern  *)
          Gen(5,Level-vlevel,DatenAdresse,'und sichern')
        END ELSE IF kind () Prozedur THEN
          BEGIN (* Charactervariable    *)
(*          UP READ_Character aufrufen  *)
            Gen(11,0,ReadChar,'Character einlesen');
(*          Character-Variable sichern  *)
            Gen(6,Level-vlevel,DatenAdresse,'und sichern');
        END ELSE error(21,'Prozeduren sind hier nicht erlaubt');
      END (* With *);
      Getsym;
    END ELSE error(14,'Nach READ muß Identifier folgen');
  END ELSE IF sym = WRITEsy THEN BEGIN GetSym;
    IF sym = semicolon THEN BEGIN
      Gen(1,0,13,'Lade 0Dh'); Gen(11,0,WriteChar,'CR schreiben');
      Gen(1,0,10,'Lade 0Ah'); Gen(11,0,WriteChar,'LF schreiben');
    END ELSE BEGIN Expression;
      IF Expression_Typ = Integer_Objekt THEN
        Gen(11,0,WriteInt,'Integer schreiben')
      ELSE Gen(11,0,WriteChar,'Character schreiben');
    END
  END ELSE ...
END (* Statement *);

(* ------------------------------------------------------- *)
(* Modul DC-006.INC (PARS-004.INC) Hilfs/Analyseproc's Parser *)
(* ------------------------------------------------------- *)
PROCEDURE Block;
...
VAR ...
    L0 : INTEGER;
... ConstDeklaration bleibt gleich ...

PROCEDURE VarDeclaration;
VAR Obj  : ObjPtr; Name : T_Name;
BEGIN Name := '';
      ...
      IF sym = Charsy THEN BEGIN Idname := Name;
        Obj := NewObj(CharVar); Obj^.VLevel := Level; GetSym;
      END ELSE IF sym = Intsy THEN BEGIN Idname := Name;
        obj := NewObj(IntVar); Obj^.VLevel := Level; GetSym;
      END ELSE error(33,'Nach ":" INT oder CHAR erwartet')
      ...
END (* VarDeclaration *);
```

```
BEGIN (* -- Block ------------------------------------------ *)
...
  IF level = 0 THEN Bottom := hd;
  DatenNr := 1; L0 := 0;
(* --- Aufbau des Activation-Records ------------------------ *)
  Gen(13,0,0,'Basepointer sichern, neuen laden');
(* --- Bearbeitung der Konstantendeklarationen -------------- *)
  ...
(* --- Bearbeitung der Prozedurdeklarationen ---------------- *)
  IF sym = PROCsy THEN BEGIN
(*  Sprung ueber den Code des(r) Unterprogramm(e) zum Anfang *)
(*  der zu diesm Block gehoerigen Statements ---------------- *)
    L0 := MakeLabel;
    Gen(9,0,0,'Mgl. Unterprogramme ueberspringen') ;
  END;
  WHILE sym = PROCsy DO BEGIN GetSym;
    ...
    obj := NewObj(Prozedur); (* Einfuegen: *)
    WITH Obj^ DO BEGIN ZWC_Startadresse := ZCode_Adresse;
      ASM_Startadresse := ASMCode_Adresse;
      COM_Startadresse := MCode_Adresse;
    END;
   (* weiter mit ... TestSemicolon; level:=level+1;...*)
(* --- Beginn der Bearbeitung des eigentlichen Blocks ----- *)
  IF sym = BEGINsy THEN Getsym ELSE error(8,'BEGIN erwartet');
(* --- Falls Prozeduren deklariert worden sind, Sprung ---- *)
(* --- vom Anfang des Blocks (ueber deren Code) eintragen - *)
  IF L0 ) 0 THEN FixUp(L0);
(* --- Analyse des Rumpfes ---------------------------------- *)
...
  IF sym = ENDsy THEN Getsym ELSE error(9,'END erwartet');
(* -- Uebriggebliebene EXIT-Spruenge als RETURN eintragen - *)
  IF ExitSprung () NIL THEN FixUpExitListe;
(* --- RETURN erzeugen nach Absprungpunkt + 1. ------------ *)
  IF Level () 0 THEN Gen(12,1,0,'Return vom Unterprogrammaufruf')
                ELSE Gen(12,0,0,'Return vom Hauptprogrammaufruf');
(* --- Naechst hoehere Symboltabelle anwaehlen, ----------- *)
...
END (* Block *);

PROCEDURE Parse;
VAR BDOS_Adr : INTEGER ABSOLUTE $0006;
BEGIN (* --- Analyse des Programmkopfes ------------------- *)
  IF sym = MODULEsy THEN BEGIN GetSym;
    IF sym = Identifier THEN BEGIN GetSym; TestSemicolon; END
    ELSE error(4,'Parse : Identifier erwartet');
  END ELSE error(1,'Parse : "MODULE" erwartet');
(* -- Basis-Zeiger fuer die Aktivierungsrecords ---- *)
(* -- auf BDOS-1 initialisieren -------------------- *)
  Gen(14,0,BDOS_Adr-1,'Stack/Basepointer initialisieren');
(* Das Hauptprogramm wird ebenfalls wie ein Unterprogramm   *)
(* aufgerufen. Anschliessend Warmstart.                     *)
  Gen(7,0,ZCode_Adresse + 2,'Aufruf des Hauptprogramms');
  Gen(9,0,0,'Warmstart'); (* Programmende                  *)
(* --- Analyse des Programmrumpfes --------------------------*)
  ...
  IF noerr THEN BEGIN
(* -- Patchen der Vorwaertsspruenge auf Maschinen-,       -- *)
(*    Zwischen- und Assemblercodedatei                     -- *)
    BackPatch; WRITELN; WRITE ('Programm fehlerfrei');
  END ELSE BEGIN
    WRITE ('Programm fehlerhaft - Fehlerzahl: ',errcount:3);
  END;
  WRITE(' - ENTER druecken');REPEAT READ(KBD,ch) UNTIL ORD(ch)=13;
END (* Parse *);

(* ------------------------------------------------------------ *)
(* DC-007.INC                    Hilfsprozeduren des Compilers *)
(*------------------------------------------------------------ *)
PROCEDURE ListMsg(drucken:BOOLEAN;s:Str255);
VAR i : INTEGER;
BEGIN
  IF drucken THEN BEGIN
    WRITELN('Drucker fertig machen und Taste druecken'); Wait;
    ClrScr; WRITELN(LST,s);
    FOR i := 1 TO Length(s) DO WRITE(LST,'-'); WRITELN(LST);
    WRITELN(LST);
  END;
  ClrScr; HighVideo; WRITELN(s);
  FOR i := 1 TO Length(s) DO WRITE('-'); WRITELN;
  NormVideo; WRITELN; WRITELN('Weiter mit (Taste)'); WRITELN;
END (* ListMsg *);

PROCEDURE ZeigeZwischencode(drucken:BOOLEAN);
VAR i : INTEGER; (* Zwischencode-Datei auf CRT zeigen *)
BEGIN ClrScr;
  IF NOT Exist(ProgrammName+'.ZWC') THEN
    WRITELN('-- Datei existiert nicht --')
  ELSE BEGIN ListMsg(drucken,
          'Adresse OpCode                Level  Adresse/Wert');
    RESET(ZWC_File); i := 0;
    REPEAT   i := i + 1; READ(ZWC_File,ZCode);
      WITH ZCode DO BEGIN
        WRITELN(nr:5,' : ',Memo[bc],lv:10,Adresszahl(ad):10:0,
                ' ; ',km);
        IF drucken THEN WRITELN(LST,nr:5,' : ',Memo[bc],lv:10,
                        Adresszahl(ad):10:0,' ; ',km);
      END;
      IF (i MOD 21 = 0) AND NOT (drucken) THEN Wait;
    UNTIL EOF(ZWC_File);
  END;
  WRITELN('----- Fertig - Taste druecken -----'); Wait;
END (* ZeigeZwischencode *);

PROCEDURE InitParser;
VAR i, ende : INTEGER; Lib : TEXT;
(*  Dateipuffer: Laenge mind. (MCodeOffset-TPA_Anfang) Byte *)
(*               hier : $200 - $100 = 256 Byte              *)
    pf      : ARRAY[1..256] OF CHAR;
BEGIN
  Jump_on_Condition := FALSE;
  ZCode_Adresse   := ZCode_Offset; (* Zwischencode-Startadresse *)
  MCode_Adresse   := MCode_Offset; (* Maschinencode-Startadresse*)
  ASMCode_Adresse := ASMCode_Offset;(*Assemblercode-Startadresse*)
  ipatch          := 1;            (* Patchfeldzeiger          *)
(* -- PatchFeld initialisieren ------------------------------ *)
  FOR i := 1 TO MaxPatch DO WITH PatchFeld[i] DO BEGIN
    Z_Adresse := 0; Z_Patch := 0; A_Adresse := 0; A_Patch := 0;
    JPC := FALSE; M_Adresse := 0; M_Patch := 0;
  END;
  TopScope := NIL;   (* Symboltabelle initialisieren *)
  ExitSprung := NIL; (* Exittabelle initialisieren   *)
  MEM[$80] := 0;     (* CP/M-Parameterzeile loeschen *)
  Noerr := FALSE;
(* -- Programmname von CRT oder Eingabepuffer holen --------- *)
  IF ProgrammName () '' THEN
    Noerr := Exist(ProgrammName+'.SRC');
  WHILE NOT Noerr DO BEGIN WRITELN; HighVideo;
    WRITELN('Bitte Sourcenamen eingeben: ((CR) = ',
            ProgrammName+'.SRC');
    NormVideo; READLN(Satz);
    FOR i := 1 TO Length(Satz) DO Satz[i] := UpCase(Satz[i]);
    IF Satz () '' THEN ProgrammName := Satz;
    Noerr := Exist(ProgrammName+'.SRC');
  END;
(* - nun werden die Compilerdateien initialisiert ---------- *)
  ASSIGN(Source,ProgrammName+'.SRC'); RESET (Source);
  Ist_Offen[SRC] := TRUE;
  IF ZWC_Gen THEN BEGIN ASSIGN (ZWC_File,ProgrammName+'.ZWC');
    REWRITE(ZWC_File); Ist_Offen[ZWC] := TRUE;
  END;
  IF COM_Gen THEN BEGIN
    IF NOT Exist('DCRTS.LIB') THEN BEGIN HighVideo;
      WRITELN('Die Datei DCRTS.LIB mit dem Laufzeitsystem ist');
      WRITELN('auf dieser Diskette.');
      WRITELN('Bitte pruefen und neu starten.'); HALT; NormVideo;
    END;
    ASSIGN(COM_File,ProgrammName+'.COM'); REWRITE(COM_File);
    Ist_Offen[COM] := TRUE;
    ASSIGN(Lib,'DCRTS.LIB'); RESET(Lib); ClrScr; HighVideo;
    WRITELN('Kopieren des Laufzeitsystems...'); NormVideo;
    ende := MCode_Offset - TPA_Anfang;
    FOR i := 1 TO ende DO READ (Lib,pf[i]);
    FOR i := 1 TO ende DO WRITE(COM_File,pf[i]);
    CLOSE(Lib);
  END;
  IF liston THEN BEGIN ASSIGN(List,ProgrammName+'.LST');
    REWRITE(List); Ist_Offen[PRN] := TRUE;
  END;
  IF ASM_Gen THEN BEGIN ASSIGN (ASM_File,ProgrammName+'.ASM');
    REWRITE(ASM_File); Ist_Offen[ASM] := TRUE;
  END;
  InitScanner; Getsym; (* -- Scanner initialisieren --------- *)
END (* InitParser *);

PROCEDURE EndParser;
BEGIN ClrScr;
  IF Ist_Offen[SRC] THEN BEGIN CLOSE(Source);
    Ist_Offen[SRC] := FALSE; END;
  IF Ist_Offen[PRN] THEN BEGIN CLOSE(List);
    Ist_Offen[PRN] := FALSE; END;
  IF (Ist_Offen[COM]) AND NOT (Noerr) THEN BEGIN CLOSE(COM_File);
    ERASE(COM_File);Ist_Offen[COM] := FALSE;END;
  IF (Ist_Offen[ZWC]) AND NOT (Noerr) THEN BEGIN CLOSE(ZWC_File);
    ERASE(ZWC_File); Ist_Offen[ZWC] := FALSE;END;
```

```
  IF (Ist_Offen[ASM]) AND NOT (Noerr) THEN BEGIN CLOSE(ASM_File);
    ERASE(ASM_File); Ist_Offen[ASM] := FALSE;END;
END (* EndParser *);

PROCEDURE Compile;
BEGIN (* Parserhauptprogramm *)
  InitParser; Parse; EndParser;
END (* Compile *);

(* ------------------------------------------------------------ *)
(* Modul DC-008.INC                                             *)
(* Hilfs- und Dialogprozeduren des Compilers                    *)
(* ------------------------------------------------------------ *)

PROCEDURE InitDC;
(* Initialisierung der Dateikontrollvariablen des Compilers  *)
BEGIN   ProgrammName := 'TEST';
  IF ParamCount ) 0 THEN ProgrammName := ParamStr(1);
  COM_Gen         := FALSE; liston         := FALSE;
  ZWC_Gen         := FALSE; ASM_Gen        := FALSE;
  Ist_Offen[SRC] := FALSE; Ist_Offen[COM] := FALSE;
  Ist_Offen[PRN] := FALSE; Ist_Offen[ZWC] := FALSE;
  Ist_Offen[ASM] := FALSE;
END (* InitDC *);

PROCEDURE SchliesseDateien;
BEGIN (* Schliesst die Ausgabedateien des Compilers          *)
  IF Ist_Offen[SRC] THEN BEGIN
    CLOSE(Source); Ist_Offen[SRC] := FALSE; END;
  IF Ist_Offen[COM] THEN BEGIN
    CLOSE(COM_File); Ist_Offen[COM] := FALSE; END;
  IF Ist_Offen[PRN] THEN BEGIN
    CLOSE(List); Ist_Offen[PRN] := FALSE; END;
  IF Ist_Offen[ZWC] THEN BEGIN
    CLOSE(ZWC_File); Ist_Offen[ZWC] := FALSE; END;
  IF Ist_Offen[ASM] THEN BEGIN
    CLOSE(ASM_File); Ist_Offen[ASM] := FALSE; END;
END (* Schliessedateien *);
```

```
PROCEDURE DisplayMenue;
(* Hauptmenue des Compilers mit den verschiedenen Optionen    *)
VAR i : INTEGER;

  PROCEDURE ZeigeFiles(s1:Str40;s2:Str4;Schalter:BOOLEAN);
  (* Zeige Arbeitsdateien *)
  BEGIN   WRITE(s1,' : ');
    IF Schalter THEN WRITE(ProgrammName + s2); WRITELN;
  END (* ZeigeFiles *);

BEGIN (* DisplayMenue *)
  ClrScr; HighVideo; ZWC Gen := TRUE;
  WRITELN('D O I T - C o m p i l e r  -  Hauptmenue');
  WRITELN('-----------------------------------');
  NormVideo; WRITELN;
  ZeigeFiles('Arbeitsdatei      ','.SRC',TRUE);
  ZeigeFiles('Zwischencodedatei ','.ZWC',ZWC_Gen); WRITELN;
  WRITELN('Compilieren_____________________(C)'); WRITELN;
  WRITELN('Compileroptionen________________(O)');
  WRITELN('neue Arbeitsdatei waehlen_______(W)');
  WRITELN('Zwischencode zeigen_____________(I)');
  WRITELN('Zwischencode drucken____________(5)'); WRITELN;
  WRITELN('Ende____________________(X) oder (Q)');
  READ(KBD,ch); ch := UpCase(ch);
  CASE ch OF
    'W'      : BEGIN SchliesseDateien; WRITELN; HighVideo;
               WRITE('Neue Arbeitsdatei: '); NormVideo;
               READLN(ProgrammName);
               FOR i := 1 TO Length(ProgrammName) DO
                ProgrammName[i] := UpCase(ProgrammName[i]);END;
    '1','5' : ZeigeZwischenCode(ch='5');
    ELSE     ;
  END (* CASE ch OF ... *);
END (* DisplayMenue *);
```

In diesem Stadium ist der DOIT-Compiler bereits in der Lage, Code für eine virtuelle Stack-Maschine zu erzeugen. ct

HW ELEKTRONIK
2000 Hamburg 19 Eimsb. Chaussee 79
Telefon 040/439 68 48 u. 430 00 19
Ihr Elektronik-Fachgeschäft in Hamburg

DIESE ANGEBOTE GELTEN AUCH IM LADENVERKAUF

TEAC-FLOPPY-LAUFWERKE
FD 35A ... 389,–
FD 35B ... 399,–
FD 35E ... 384,–
FD 35F ... 425,–
FD 35F-PS ... 440,–
FD 55A-V ... 349,–
FD 55B-V ... 344,–
FD 55E-V ... 399,–
FD 55F-V ... 369,–
FD 55GF-V ... 429,–
SD 510 Winch., 10 MB ... 1499,–
SD 520 Winch., 20 MB ... 1899,–

NEC-FLOPPY-LAUFWERKE
FD 1053 ... 399,–
FD 1055 ... 409,–
FD 1155 C ... 475,–
FD 1035 ... 415,–
D 3126 ... 2089,–
D 5124, 10 MB ... 1859,–
D 5126, 20 MB ... 1999,–
D 5146,40 MB ... 3269,–

DEN müssen Sie haben:
HW-KATALOG 1986
(mit Bestellung gratis, oder Porto DM 1,40)

3M-Disketten — 10 St.
3,5" SS/DD ... 79,50
3,5" DS/DD ... 108,50
5,25" SS/DD ... 39,30
5,25" DS/DD ... 51,50
5,25" SS/DD 96 TPI ... 54,20
5,25" DS/DD 96 TPI ... 64,90
5,25" DS/QD 96 TPI ... 111,20
7740 Reinig.-Set ... 43,–

Exp.-Karte f. „IBM-PC" o. ä.
Epoxyd, ca. 4 k Löcher ... 45,–

EXP.-KARTE f. „APPLE" o.ä.
2seitig, Epoxyd ... 12,50

EPROM-LÖSCHGERÄT
m. Timer
max. 6 EPROMS/12 V = 86,90

QUALITÄTS-QUARZE
32,768 kHz ... 1,75
100 kHz ... 19,50
1,0/1,008/1,8432 ... je 8,30
2,0/2,048/2,097152/
2,4576 MHz ... je 5,20
3,0/3,2768/3,579545/
4,0/4,096/4,194304/
4,433618/4,9152/5,0/
5,0688/5,12/5,185/
5,2/6,0/6,4/6,5536/
7,3728/8,0/8,877238/
9,216/10,0 MHz ... je 2,20
10,24/12,0/14,0/14,318/
14,7456/15,0/16,0/
18,0 MHz ... je 2,40

QUARZ-OSZILLAT
4/5/6/8/10/12/14/16/18/
24 MHz, ... je 8,90

TEXTOOL-TESTSOCKEL
16polig ... 20,50
20polig ... 22,–
24polig ... 21,25
28polig ... 22,40
40polig ... 31,20
64polig ... 125,–

ECB-Bus
:0 St.-pl. ... 36,50
DTO, m. 10 BL 64p ... 65,90

IC-QUALITÄTSSOCKEL DOPPELFEDER
8–40polig p.P. ... 0,02

PRÄZISIONSSOCKEL vergold., gedreht
8–64p p.P. ... 0,05

PRÄZISIONSSOCKEL WIRE-WRAP/ verg., gedreht
8–40p p.P. ... 0,20

DIL-STECKER
f. Flb-KABEL/anpreßb.
DIS 14 1,50 DIS 28 2,20
DIS 16 1,65 DIS 40 2,80
DIS 24 2,–

FLOPPY-STECKER verg.
10polig ... 3,50
20polig ... 4,95
26polig ... 5,50
34polig ... 5,95
40polig ... 7,25
50polig ... 8,90
60polig ... 11,50

CENTRONICS-STECKER verg.
14polig/löt ... 5,50
24polig/löt ... 7,50
24polig/quetsch ... 11,50
36polig/löt ... 5,90
36polig/quetsch ... 8,50
50polig/löt ... 10,50
50polig/quetsch ... 17,90

CENTRONICS-BUCHSEN verg.
14polig/löt ... 5,50
24polig/löt ... 7,90
24polig/quetsch ... 11,50
24polig/90° ... 9,50
36polig/löt ... 5,90
36polig/quetsch ... 8,50
36polig/90° ... 9,90
50polig/löt ... 10,50
50polig/quetsch ... 19,50

CENTRONICS-KUPPLUNGEN
36polig/löt ... 9,90
50polig/löt ... 16,90

VG-STECKVERBINDER verg.
64p Stecker A/C ... 2,20
96p Stecker ... 4,80
64p Buchse A/C ... 2,85
64p Buchse 90° ... 7,90
64p Buchse Wire Wrap ... 4,80
64p Buchse/quetsch ... 15,–
96p Buchse ... 5,20
96p Buchse Wire Wrap ... 6,95

DIREKTSTECKER/SLOTS verg.
2× 18p RM 2,54 ... 5,50
2× 22p dto ... 5,90
2× 23p dto ... 6,20
2× 25p dto ... 4,95
2× 31p dto ... 7,95
2× 6p RM 3,96 ... 2,95
2× 12p dto ... 4,50
2× 18p dto ... 6,90
2× 22p dto ... 7,20

SUB-D-STECKVERBINDER verg.
STECKER

Pole	löt	90°	quetsch
9	1,55	3,90	7,70
15	1,75	5,40	9,90
25	2,50	7,90	9,75
37	4,50	11,50	15,90
50	5,–	15,50	39,–

BUCHSEN

9	2,–	3,95	8,20
15	2,50	5,50	9,90
25	3,–	8,75	9,75
37	5,20	13,90	14,90
50	7,20	16,50	49,–

Printerkabel f. IBM o. ä.
2 m/Centr.-St. 36p auf
Sub-D-St. 25p ... 32,50

SUB „D"-GEHÄUSE

Geh.	oberh.	seitl.
9	2,20	2,70
15	2,50	2,95
25	2,60	3,20
37	3,10	3,70
50	3,50	3,90

PFOSTENFELD-VERBINDER
trennbar
36p Stecker 1reihig ... 2,20
36p Stecker 90° ... 2,75
72p Stecker 2reihig ... 4,50
72p Stecker 90° ... 4,95
36p Buchse 1reihig ... 3,95
72p Buchse 2reihig ... 7,50
36p Buchse 1reihig 90° ... 4,75
72p Buchse 2reihig 90° ... 9,50
JUMPER rt/sw ... 10 St. 2,20

PFOSTEN-STECKVERB.
vergold. m. Zugentlastg./
Auswerfer/2reihig

STECKER-/90°		BUCHSEN
10p 1,60	2,65	1,95
14p 1,60	2,75	2,50
16p 1,80	2,90	2,40
20p 2,60	3,30	3,25
26p 2,70	3,95	3,50
34p 3,60	5,20	3,75
40p 4,20	5,75	5,20
50p 5,00	7,25	6,50
60p 8,75	8,20	7,50

FLOPPY-STROM-VERSORGUNGEN 4p
Stecker (3,5") ... 1,70
Stecker (5,25") ... 1,70
Kuppl. (3,5") ... 1,70
Kuppl. (5,25") ... 1,70
Platinenst. (3,5") ... 1,70
Platinenst. (5,25") ... 1,70
Stecker, 6polig ... 2,75
Kuppl., 6polig ... 2,75

Widerstands-Netzwerke
Reihe E12: 100 bis 1M
SIL 7 7 × R gem. Masse ... je 0,95
SIL 8 8 × R gem. Masse ... je 1,25
SIL 9 9 × R gem. Masse ... je 1,50

74 ALS. 74 ALS. 74 ALS.
00/01/02/03/04/05/08/09/
10/11/20/21/22/27/28/30/
32/33/37/38/40 ... je 1,20
15/74/86/109/112/113/114/
133 ... je 1,50
137/138/151/153/160/161/
162/163/174/175/193/251/
253/257 ... je 1,85
168/169/191/192/240/241/
243/244/273/373/374 ... je 2,50
540/541/640/641/642/643/
644/645 ... je 3,00
245/804/805/806/832 ... je 3,50

74 LS. 74 LS. 74 LS. 74 LS.
00/02/03 ... je 0,55
01/04/05/08/09/10/11/12/
51/54/55/133/260 ... je 0,65
13/15/20/21/22/26/28/30/
32/33/37/38/136 ... je 0,70
14/40/73/74/75/76/
78/107/109/112/113/
114/122/266/365/
366/367/368/386 ... je 0,95
42/86/90/91/92/93/
125/126/132/137/
138/139/279/375 ... je 1,20
48/49/83/85/95/96/
123/151/153/155/
156/157/158/160/
161/162/163/164/
173/174/175/193/
194/195/196/197/
251/253/257/258/
283/290/293/352/
353/390/393 ... je 1,40
47/145/169/170/190/
191/192/221/240/
241/242/243/244/
247/248/249/259/
273/295/298/373/
374/377/378/379/
395/396/490/540/
541/640/641/642/
643/644/645/668/
669 ... je 1,70
165/166/245/280/
348 ... je 1,95
63/147/148/398/
399/670 ... je 2,40
154/183/624/626 ... je 3,50
275/321/323 ... je 5,20
673/674/682/683/
685/688/689 ... je 6,30
686/687 ... je 7,90
590/592 ... je 9,90
Serien 74HC, 74S, 74AS, C/M 4000, 74 HCT preiswert ab Lager!

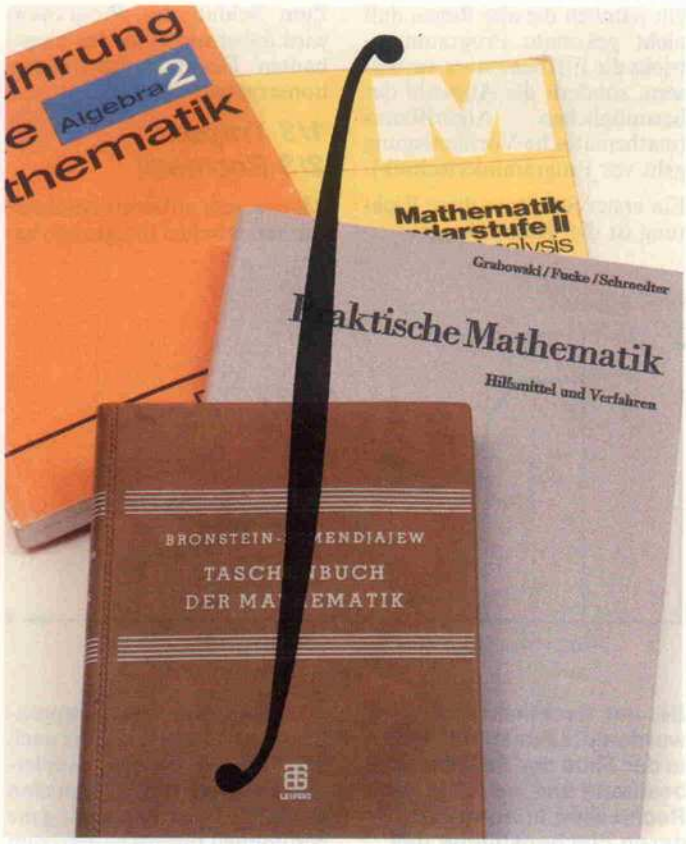
* µP µP µP µP µP *

2532	12,90
2564-450	28,90
2716-350	9,50
2716-450	7,90
2732A 250	8,–
2764-250	5,90
27C64-250	10,50
27128-250	7,90
27256-250	12,90
2114L200	5,90
4116-200	3,40
4164-120 (128Z)	4,95
4164-150	3,95
4164-150 autom.	4,95
41256-120	12,50
41256-150	8,75
6116LP3	5,50
6264LP15	8,90
6502	9,90
6502A	11,90
65C02	18,90
65SC02	22,–
65SC02-2	24,90
6504	12,95
6504A	14,20
65SC04	19,50
65SC04 2	20,90
6510	49,–
6520	7,90
6520A	9,50
65SC21	17,50
65SC21 2	19,50
6522	9,90
6522A	10,90
65SC22	19,40
65SC22-2	24,75
6526	49,50
6532	12,50
6532A	16,90
65SC32	22,50
65SC32-2	24,50
6545-1	19,90
6551	12,90
6551A	13,50
65SC51	19,40
65SC51-2	24,50
6568	139,–
6581	77,–
6802	8,50
68A02	11,50
6808	14,–
6809	19,–
68A09	22,–
6810	6,50
6821	3,90
68A21	4,20
6845	10,90
6850	4,75
68000-8	49,50
68008-8	53,90
68010L8	169,–
7910 Mod.	78,90
7911 Mod.	78,90
8080A	6,90
8085A	7,–
8086	23,–
8087 2	529,–
8088	19,50
8155	7,50
8216	4,50
8224	4,50
8237	17,50
8243	5,90
8251	5,90
8253	5,90
8255	4,90
8257	7,40
8259	7,75
8279	8,25
8282	7,50
8284	8,50
8286	7,70
8288	18,90
AM25LS2538	8,50
AM26LS31	6,95
AM26LS32	6,95
AM26LS33	6,95
AY 3 1015	12,40
AY 3 8910	19,–
AY 3 8912	17,50
EF 9364	17,50
EF 9365, 66	je 69,–
EF 9367	74,50
FDC 9229 B	53,50
HD 4702	23,50
ICL 7660	8,50
N 82 S 23	5,50
N 82 S 123	5,50
N 82 S 126	6,50
N 82 S 129	6,50
RTC 58321	17,90
TMS 4416-15	10,90
TMS 4500-19	49,–
TMS 9902	19,90
TMS 9995	64,–
1791	21,90
1793	21,90
1797	21,90
2143	22,90
2797, 2793	je 27,50
9216	15,90
Z 80A C/MOS	14,90
Z 80A CPU	4,25
Z 80A PIO	4,95
Z 80A CTC	4,95
Z 80A DART	13,50
Z 80A DMA	12,90
Z 80A SIO-O	13,90
Z 80A STI	45,–
Z 80B CPU	6,90
Z 80B PIO	9,50
Z 80B CTC	9,50
Z 80B DART	19,20
Z 80B SIO-O	20,90
Z 80B STI	44,50
Z 80H CPU	10,90
ZN 416	6,50
ZN 425	13,50
ZN 426	9,70
ZN 427	22,–
ZN 428	21,50
µPD 765	13,90
µPD 1990	9,50
µPD 70108D8	35,–
µPD 70116D8	38,90
µPD 71055C	8,50
µPD 71088C	23,90
µPD 7220AD	39,90

KONDENSATOREN 100 nF/RM 5 mm KERAMIK
100 St. ... 21,–
500 St. ... 95,–

GEFRAGTES BEI HW!!
TC 51100 C 12 ... 329,–
7406, 07 ... je 1,60
MC 1488, 89 ... je 1,80
2149-2 ... 12,50
8 T 97 ... 2,95
AY-5-8100 ... 5,–
DM 81 LS 97 ... 4,50
MM 5204Q ... 3,–
68000-12 ... 119,–
Z 80 DMA ... 7,50
Z 80 SIO-1 ... 9,50
TL 7705 ... 3,30

PREISE INKL. MWST.
Versand ab DM 30,– ausschl. per Nachnahme!
Zwischenverk. vorbeh.
Erfüllungsort: Hamburg



Integrieren lassen

Numerische Integration in BASIC

Frank Schmidt

Dies soll kein Aufruf an angehende Abiturienten sein, ihre Hausaufgaben heute mal bleiben zu lassen. Auch ist es nicht Ziel und Zweck dieses Artikels, zu zeigen, wie man sich diese vom Computer abnehmen läßt. Vielmehr stellt der Artikel dar, wie sich ein Rechner zur Lösung mathematischer Probleme einsetzen läßt, an die man ohne ihn nicht oder nur sehr mühsam herankommen kann.

In der Mathematik und erst recht in der Physik treten immer wieder Probleme auf, die sich nicht exakt mathematisch lösen lassen. Hierzu gehört unter anderem die Nullstellenbestimmung bei Polynomen, deren Grad drei übersteigt, aber auch die Integration selbst 'einfach' anmutender Funktionen wie zum Beispiel $f(x) = \sin(x)/x$, welche nicht mehr elementar durchführbar ist.

Es gibt aber Methoden, um solche Integrale wenigstens numerisch, das heißt näherungsweise ihrem Zahlenwert nach zu bestimmen. Und welches Hilfsmittel bietet sich dabei eher an als ein Computer.

Die angeführten Beispielprogramme sind in BASIC geschrieben, der Allgemeinverständlichkeit zuliebe. Wie man vielleicht erkennen kann, wurde aber nicht 'irgendein' BASIC benutzt, sondern der (Pseudo-) Compiler von Digital Research. Dieser besitzt nämlich

- gute Strukturierbarkeit und
- sehr gute Rechengenauigkeit.

Vor allem auf die Rechengenauigkeit kommt es natürlich bei diesen numerischen Berechnungen an.

Mit Sehnen ...

Als erstes Integrationsbeispiel kommt die Funktion $f(x) = 1/x$ (eine Hyperbel) zum Einsatz. Hier liefert – wie man vielleicht noch aus der Schule weiß – der natürliche Logarithmus ($\ln(x)$) das Ergebnis. Integriert werden soll der Einfachheit halber in den Grenzen (1,a). Die Methode, die hier zur Anwendung kommt, ist die Sehnentrapezformel.

$$F(x) = \sum_{i=0}^{n-1} \frac{h}{2} \left(f(x_i) + f(x_{i+1}) \right) + e$$

Dabei wird die Fläche unter der Funktion f in einzelne Trapeze aufgeteilt. Die gemeinsame Breite dieser Trapeze wird mit h bezeichnet. Die oberen Eckpunkte bilden jeweils zwei Funktionswerte. Dieses Prinzip, die Randpunkte jedes Intervalls als Stützstellen zu benutzen, trägt in der Literatur die Bezeichnung 'Prinzip von Newton-Kotes'.

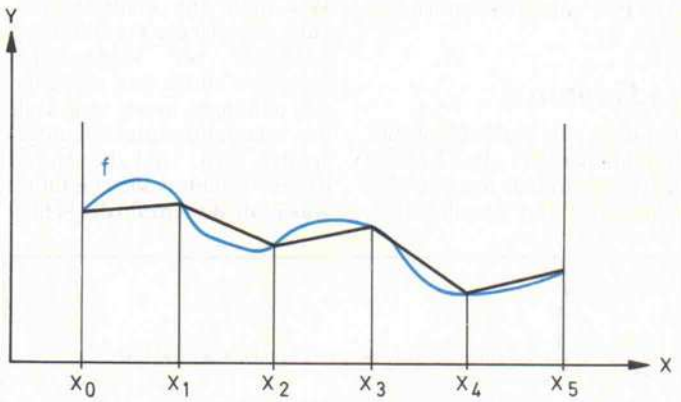
Der sogenannte Verfahrensfehler 'e' bei dieser numerischen Integration kann auch mathematisch abgeschätzt werden:

$$|e| \leq (b - a) \frac{h^2}{12} \max |f''(x)|$$

Er hängt in erster Linie ab von der zweiten Ableitung f''(x), also der Krümmung der Funktion f, und von der Breite h der Trapeze beziehungsweise der Anzahl n der Unterteilungen des Intervalls. Vereinfacht man die Summe etwas (bis auf den Anfangs- und Endwert kommt ja jede sogenannte Stützstelle f(x) zweimal vor), so erhält man die vereinfachte Formel:

$$F(x) = h \left(\frac{1}{2} f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2} f(x_n) \right) + e$$

nach der auch das Beispielprogramm arbeitet. Dabei wird hier aber nicht die Anzahl n der Intervallunterteilungen, sondern gleich die gewünschte Genauigkeit eingegeben und n daraus



So funktioniert das Sehnentrapezverfahren: Die zu integrierende Funktion ist blau dargestellt, die bei der Integration berechneten Trapeze schwarz. Die Funktionswerte werden jedesmal an den Rändern der Teilintervalle bestimmt und stellen die Ecken der Trapeze dar.

nach der obigen Abschätzung berechnet. Das hat den Vorteil, daß man sofort weiß, wie exakt der berechnete Wert nun wirklich ist.

Um nachzuprüfen, ob die geforderte Genauigkeit auch wirklich eingehalten wurde, berechnet der Computer den gesuchten Funktionswert noch einmal mit Hilfe der 'eingebauten' Funktion LOG und teilt dann auch die Abweichung von dem durch Integration ermittelten Wert mit.

Ein paar Worte noch zu einigen

Feinheiten in diesem Programm:

Als erstes achtet es darauf, daß die eingegebenen Integrationsgrenzen echt größer als Null sind, da für andere Werte der Logarithmus nicht definiert ist (zumindest im Reellen). Falls nötig, werden sodann die beiden Grenzen ausgetauscht, damit auch wirklich a die kleinere ist und die Integration ohne Schwierigkeiten von a nach b ausgeführt werden kann. Schließlich wird das Maximum m der zweiten Ableitung der Funktion berechnet. Die Ableitung lautet

$f''(x) = 2x^{-3}$

und ist natürlich da am größten, wo x selbst am kleinsten ist, also an der Stelle a. Zum Schluß wird dann die Summe ausgerechnet und mit dem richtigen Vorzeichen versehen (wurden die Integrationsgrenzen vertauscht, so ist jetzt das Vorzeichen umzudrehen). Beim Probelauf fällt auf, daß die Abweichung vom Sollwert wesentlich kleiner ist, als die vorgegebene Fehlergrenze e. Das liegt daran, daß die Abschätzungsformel den maximalen Wert für e liefert. In Wirklichkeit ist dieser aber in der Regel kleiner. Nur weiß man halt nicht genau, wie klein er ist und hat deshalb den schlimmsten Fall anzunehmen (worst case).

. . . Fehlern . . .

Hier noch eine Bemerkung zum Verfahrensfehler: Es handelt sich hier wirklich nur um den Fehler, den das jeweilige Rechenverfahren in sich trägt. Dieser Verfahrensfehler ist nicht notwendigerweise identisch mit der Abweichung des vom Computer 'ausgespuckten' Ergebnisses vom wirklichen Wert, denn der Rechner fügt dazu noch seine eigenen Rechenungenauigkeiten, also Rundungsfehler, hinzu.

So ist es durchaus möglich, daß man ein Ergebnis erhält, welches sich ausschließlich aus solchen Rechenungenauigkeiten zusammensetzt! Bei jeder Addition muß man daher einen Fehler in der Größenordnung der letzten Stelle der Summanden mit einkalkulieren, was sich natürlich bei CBASIC, welches mit 14 Stellen Genauigkeit (und BCD-Kodierung) rechnet, nicht sonderlich auswirkt. Benutzt man allerdings einen BASIC-Interpreter, der nur mit sechs Stellen arbeitet, so muß man nach 1000 Additionen schon mit einem Fehler in der dritten Stelle rechnen! Es ist deshalb empfehlenswert, sich nach Möglichkeit um solche Additionen zu drücken. Das ist auch der Grund dafür, weshalb im Programm 1 in der FOR-NEXT-Schleife x nicht in der Form 'FOR x=a TO b STEP l', sondern als Produkt berechnet wurde.

Wie man aus Fehlerabschätzung sieht, ist die Verfahrensgenauigkeit bei vorgegebener Funktion einzig und allein davon abhängig, in wie viele Teile das Integrationsintervall aufgespalten wird. Und die Anzahl dieser Teilintervalle bestimmt wiederum die Anzahl der Schleifendurchläufe und damit letztendlich auch die Rechengeschwindigkeit.

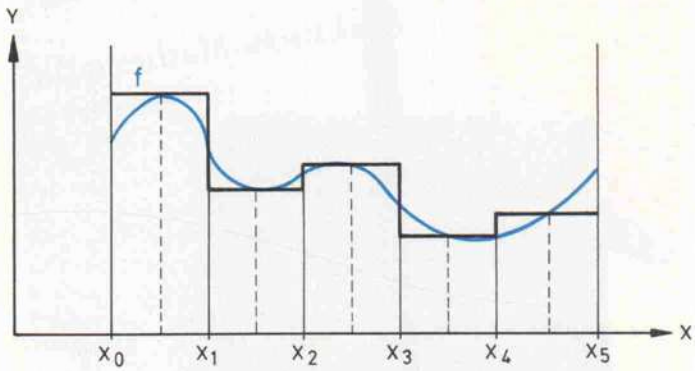
```
C>CRUN LN


CRUN VER 2.37
Bitte das Argument angeben, dessen Logarithmus bestimmt
werden soll:  b =  3

Bitte die Genauigkeit der Integration vorgeben: e =  1e-4

Die vorgegebene Genauigkeit erfordert  116 Schleifendurchläufe!
Bitte warten.
Berechneter Funktionswert:  1.09863430758
Wirklicher  Funktionswert:  1.09861228867

Differenz:  2.20189143E-05
```

Ein Probelauf von Programm 1 (Integration von 1/x). Man sieht, daß die erreichte Genauigkeit deutlich über der geforderten liegt.

. . .Rechtecken

Um bei gleicher Abweichung des Ergebnisses trotzdem die Rechenzeit zu verkürzen, muß man sich also eine raffiniertere Methode überlegen. Auch hier gilt nämlich die alte Regel, daß nicht gekonnte Programmiertricks die Effizienz stark verbessern, sondern die Auswahl des bestmöglichen Algorithmus (mathematische Vorüberlegung geht vor Programmiertechnik).

Ein erster Schritt in diese Richtung ist die 'Rechteckregel'



Bei der Rechteckregel werden die Funktionswerte in der Mitte der Teilintervalle bestimmt und als Höhe von Rechtecken interpretiert, deren Flächensumme das Integral darstellen soll.

Hier werden als sogenannte 'Stützstellen' (die Stellen, an denen der wirkliche Funktionswert berechnet wird) nicht die Ränder der Teilintervalle, sondern ihre Mittelpunkte benutzt (MacLaurin-Prinzip):

$$F(x) = h \sum_{i=1}^{n} f\left(\frac{x_{i-1} + x_i}{2}\right) + e$$

Für die Fehlerabschätzung gilt nun:

$$|e| \leq (b - a)\,\frac{h^2}{24}\,\max|f''(x)|$$

Das bedeutet schon eine Steigerung gegenüber vorher (wenn auch noch keine entscheidende). Programm 2 benutzt diese Rechteckregel, um die Funktion 'Arcus Tangens' zu simulieren.

Mit 'simulieren' ist in diesem Zusammenhang gemeint, daß sich diese (im BASIC vorhandene) Funktion als – numerisches berechnetes – Integral der Funktion

$$f(x) = \frac{1}{1 + x^2}$$

ergibt.

Zum Schluß des Programms wird daher mit Hilfe der 'eingebauten' Funktion das Integrationsergebnis überprüft.

1/3 Trapez + 2/3 Rechteck

Als eine sehr effiziente Methode zur numerischen Integration hat sich die sogenannte 'Simpsonsche Formel' erwiesen, die auch den lustigen Namen 'Keplersche Faßregel' trägt. Wenn man allerdings Funktionen mit ganz bestimmten Eigenschaften (zum Beispiel Periodizität, Monotonie und so weiter) integrieren will, kann man auf noch schnellere Verfahren zurückgreifen. Die Simpsonsche Methode ist zwar etwas umständlicher als die beiden vorher angeführten Verfahren, sie liefert aber einen kleineren Verfahrensfehler, so daß sie schneller zum Ziel führt.

$$|e| \leq (b - a)\,\frac{h^4}{2880}\,\max|f^{(4)}(x)|$$

Die Formel für die Integration lautet folgendermaßen: 'Man berechne das Integral nach der Sehnentrapezformel, addiere dazu zweimal das Ergebnis nach der Rechteckformel und teile die Summe durch drei'. Wer hätte gedacht, daß das Kochrezept so einfach wäre? Es handelt sich also schlichtweg um das gewichtete Mittel der beiden anderen Methoden. Programm 3 benutzt die Simpson-Formel, um den 'Areatangens' zu simulieren, eine Funktion, die auch als 'Arcus Tangens Hyperbolicus' bekannt ist. Diese Funktion haben zwar fast alle besseren Taschenrechner aufzuweisen, in BASIC dürfte sie aber nur in Ausnahmefällen als Standard-Funktion zur Verfügung stehen. Das macht aber nichts, da sich der 'Areatangens' auch mit Hilfe des Logarithmus darstel-

len läßt, wie man am Ende des Programms sieht.

Die so kompliziert klingende Funktion hat nun eine recht einfache Ableitung, die nach der Simpson-Formel zu integrieren ist:

$$f(x) = \frac{1}{1 - x^2}$$

Leider bereitet hier nun das Berechnen und Abschätzen der – mittlerweile schon vierten – Ableitung etwas Mühe, obgleich es sich hier nur um eine einfache, gebrochen-rationale Funktion handelt. Aber diese Berechnung braucht man ja nur ein einziges Mal (pro Funktion) auszuführen - wenn auch ohne elektronische Hilfe. . .

Integralsinus

Bis jetzt dienten die Methoden der numerischen Integration nur dazu, Ergebnisse zu beschaffen, an die man auch einfacher hätten kommen können. Zum Schluß aber folgt noch ein 'sinnvolles' Beispiel. Es soll nämlich die eingangs schon angesprochene Funktion

$f(x) = \sin(x)/x$

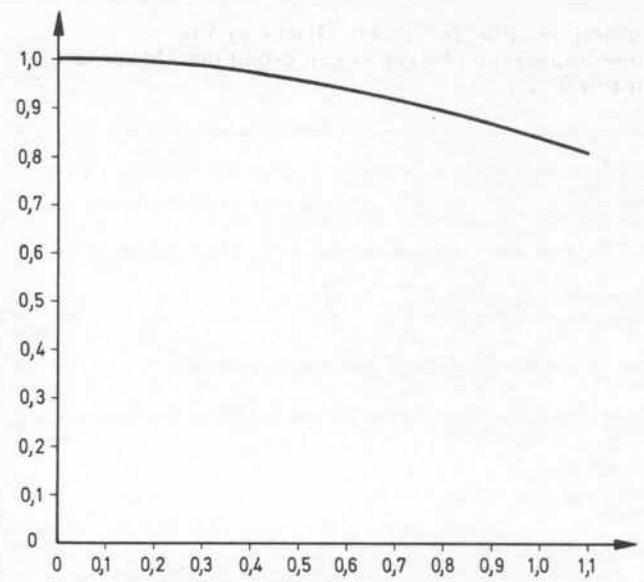
integriert werden, der sogenannte Integralsinus. Diese Funktion scheint im Nullpunkt nicht definiert zu sein, ihr Grenzwert existiert aber sehr wohl – und ist ganz einfach 1. Auch ansonsten ist die Funktion gar nicht so wild, wie es vielleicht den Anschein hat.

(–Bild 4–)

Programm 4 erstellt eine Tabelle von Integralsinus-Werten. Um einen 'Division by zero'-Error zu umgehen, wird die Rechteckregel benutzt, die den Funktionswert im Nullpunkt ja nicht benötigt. Leider lassen sich nun die vom Rechner produzierten Ergebnisse nicht mehr so einfach überprüfen, da bisher noch in keinem BASIC eine Funktion eingebaut ist, mit deren Hilfe man den Integralsinus simulieren könnte. Einem Vergleich mit in mathematischen Tafelwerken tabellierten Werten halten sie allerdings (innerhalb der entsprechenden Fehlergrenzen) ohne weiteres stand.

More BASIC

Die vorgestellten Verfahren lassen sich problemlos auf alle gängigen Microsoft-Dialekte übertragen. Es ist dazu lediglich notwendig, Zeilennummern einzufügen und den Dezimalpunkt aus der Funktionsdefinition zu entfernen. Bei MBASIC zeigte sich, daß die Ergebnisse so genau waren, daß Abweichungen vom 'eingebauten' Wert erst in den nicht mehr angezeigten Stellen auftraten. Dabei war die Rechengeschwindigkeit von MBASIC um den Faktor 4,5 größer als die von CBASIC (Programmbeispiel 1)! Im Rahmen der sechs angezeigten Stellen liefert also auch MBASIC beachtliche Ergebnisse.



So 'uninteressant' ist die Funktion sin (x)/x im Intervall [0,1]. Trotzdem läßt sie sich nicht elementar integrieren.

```
REM Numerische Bestimmung des natürlichen Logerithmus durch
REM Integration der Funktion f(x) = 1/x in den Grenzen (1,b)
REM Ergebnis : s = ln(b)

REM Methode: Sehnentrapezformel


DEF FN.f(x) = 1/x


PRINT "Bitte das Argument angeben, dessen Logarithmus bestimmt"
INPUT "werden soll:  b = "; arg
WHILE arg <= 0
   INPUT "Es muß gelten: b > 0. Neue Eingabe: b = "; arg
WEND
IF arg > 1 THEN a = 1: b = arg ELSE a = arg: b = 1

REM Berechnung des Maximums der zweiten Ableitung
m = ABS(2/(a*a*a))

INPUT "Bitte die Genauigkeit der Integration vorgeben: e = "; e
e = ABS(e)
lx = b-a
n = SQR(lx*lx*lx*m/(12*e))
IF n > 1 THEN n = INT(n)+1 ELSE n = 2
h = lx/n

PRINT "Die vorgegebene Genauigkeit erfordert "; n;
PRINT "Schleifendurchläufe!"
PRINT "Bitte warten."

s = .5 * (FN.f(a)+FN.f(b))

FOR i = 1 TO n-1
    s = s + FN.f(a+i*h)
NEXT
s = s * h
IF arg < 1 THEN s = -s

PRINT "Berechneter Funktionswert: "; s
PRINT "Wirklicher  Funktionswert: "; LOG(arg)
PRINT
PRINT "Differenz: "; ABS(s-LOG(arg))

END
```

Programm 1. Mit der Sehnentrapezformel ist das numerische Integrieren kein Problem.

```
REM Numerische Bestimmung des Arcus Tangens durch Integration
REM der Funktion f(x) = 1/(1+x*x) in den Grenzen (0,b)
REM Ergebnis : s = arctan(b)

REM Methode: Rechteckregel


DEF FN.f(x) = 1/(1+x*x)


PRINT "Bitte das Argument angeben, dessen Arcus Tangens bestimmt"
INPUT "werden soll:  b = "; arg
IF arg > 0 THEN a = 0: b = arg ELSE a = arg: b = 0

REM Berechnung des Maximums der zweiten Ableitung
m = 8

INPUT "Bitte die Genauigkeit der Integration vorgeben: e = "; e
e = ABS(e)
lx = b-a
n = INT(SQR(lx*lx*lx*m/(24*e)))+1
h = lx/n

PRINT "Die vorgegebene Genauigkeit erfordert "; n;
PRINT "Schleifendurchläufe!"
PRINT "Bitte warten."

s = 0

FOR i = 1 TO n
    s = s + FN.f(a+(i-.5)*h)
NEXT
s = s * h
IF arg < 0 THEN s = -s

PRINT "Berechneter Funktionswert: "; s
PRINT "Wirklicher  Funktionswert: "; ATN(arg)
PRINT
PRINT "Differenz: "; ABS(s-ATN(arg))

END
```

Programm 2. Bei der Rechteckregel ist der maximale Fehler nur halb so groß wie bei den Sehnentrapezen.

hardware
software
organisation
service
CSF GmbH

Heeper Str. 106–108, 4800 Bielefeld 1, 0521/61663

Kein Kabelsalat mehr mit dem Gehäuse für ATARI ST

DM 198,–
inkl. MwSt.

- Zentrale Stromversorgung für alle Geräte einschl. 2 Drucker
- Einbaumöglichkeit von 2 Diskettenlaufwerken
- Rechner (Tastatur) kann komplett unter das Gehäuse geschoben werden (Staubschutz)
- Massives Blechgehäuse

ATARI ST-Gehäuse erhalten Sie bei den autorisierten Fachhändlern

LECH-TECHNICS

Gesellschaft zur Herstellung und Vertrieb von elektrischen Geräten und Microcomputern mbH

Heerstraße 96
5014 Kerpen-Türnich
West-Germany

Telefon: 02237/8171 u. 1709
Telex: 889103 wer d

Microcomputer „ATLAS 16" wie nebenstehend, jedoch mit einem zweiten Diskettenlaufwerk **Preis: 2199,— DM**

Motherboard ohne RAM	399,— DM
Schaltnetzteil 135 Watt	299,— DM
Color-Graphik-Card	199,— DM
Monochrome-Graphik-Printer-Karte (Hercules kompat.)	299,— DM
Multifunktionskarte 384k	330,— DM
Diskdrive 2 x 40 Track	399,— DM
Multi-I/O-Karte	390,— DM
Tastatur für IBM ascii/deut	248,— DM
Disk-Controller-Karte	125,— DM
Monochrome Monitor 22 MHz	348,— DM
Monochrome Monitor TTL	448,— DM
R-G-B Farbmonitor SAKATA	1298,— DM
Harddisk 20 MB m. Contr.	1999,— DM
MS-DOS 2.11 m. GWBasic	299,— DM
Drucker SAKATA f. IBM	898,— DM
Supermodem Hayes komp.	448,— DM
Math. Co-Prozessor 8087	498,— DM
Epromer 2716/32/64/128	448,— DM
TRANSNET Netzwerkkarte	1198,— DM
IBM Metallgehäuse klappb.	128,— DM

MICROCOMPUTER „ATLAS 5"
voll APPLE IIe kompatibel

Betriebsbereit im IBM-ähnlichen Gehäuse, 128K RAM, CPU 6502A, 80- und 40-Zeichen-Darstellung (80-Zeichen-Karte serienmäßig), 7 APPLE II kompat. Slot, Schaltnetzteil 5A, Metallgehäuse, Anschlüsse für Monitor und Kassette, abgesetzte Tastatur mit Funktionstasten und 10er-Block. **Preis: 999,— DM**

Technische Änderungen vorbehalten. Endpreise zzgl. Porto und Verpackung. Ausführliche Info und Preise gegen DIN-A5-Freiumschlag und 1,30 DM Rückporto.

Besuchen Sie uns auf der „HOBBYTRONIC 86" in Dortmund vom 23.—27. 4. 1986.

MICROCOMPUTER „ATLAS 16"
voll IBM XT-kompatibel

256K-RAM, aufrüstbar bis 640K auf Motherboard, 1 Diskettenlaufwerk 360K, separate deutsche Tastatur mit 10 Funktionstasten, Schaltnetzteil 135 Watt (geeignet für Harddiskbetrieb), XT-Hauptplatine mit 8 Erweiterungsbuchsen für Zusatzkarten, CPU 8088, Taktfrequenz 4,77 MHz, 28-Pin-Sockel für ROM-Speicher (Eprom 2764), eingebautes BIOS-Programm, Color-Graphik-Karte zum Anschluß eines Monochrome-Monitors und eines R-G-B-Farbmonitors, CRT-Controller 6845. Anschlußmöglichkeit eines Lichtgriffels. Multifunktionskarte: batteriegepufferte Echtzeituhr, paralleler Ausgabeport (Centronic) zum Anschluß eines Matrixdruckers, ein asynchroner serieller Aus-/Eingabe-Communicationsport (RS 232 Interface), Möglichkeit zur Erweiterung für einen zweiten asynchronen seriellen Aus-/Eingabe-Port, Diskcontroller zum Anschluß von zwei Diskettenlaufwerken, englische Handbücher sind im Preis enthalten. **Preis: 1799,— DM**

MICROCOMPUTER „ATLAS AT"
voll IBM AT-kompatibel

640K RAM, ein Diskettenlaufwerk 1,2 MByte, eine Harddisk 20 MByte, separate deutsche Tastatur, Schaltnetzteil 192 Watt, Hauptplatine mit 8 Erweiterungsbuchsen für Zusatzkarten, CPU 80286, eingebautes BIOS-Programm, Color-Graphik-Karte zum Anschluß eines Monochrome-Monitors und eines R-G-B-Farbmonitors, Controller-Karte zum Anschluß von zwei Harddisk und zwei 1,2-MByte-Laufwerken, englische Handbücher sind im Preis enthalten. **Preis: 7699,— DM**
Ohne Harddisk **Preis: 5499,— DM**

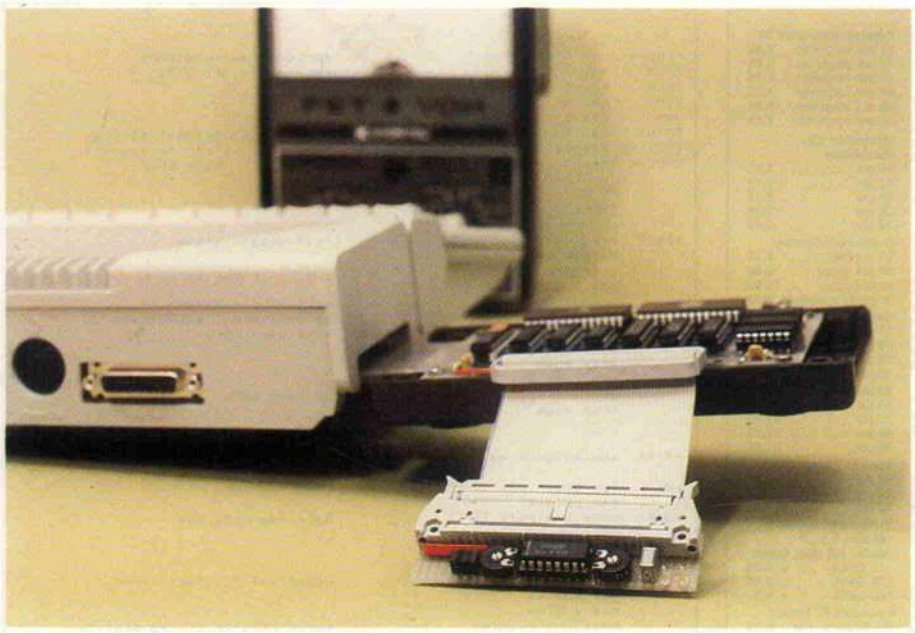
Riesen-Auswahl auf 250 m² **Kompatibles zum kleinen Preis**
(0431) 567041 die Welt der COMPUTER mcc Computer-Markt und Fachwissen am Platz Kiel

```
REM Numerische Bestimmung des Areatangens durch Integration
REM der Funktion f(x) = 1/(1-x*x) in den Grenzen (0,b)
REM mit -1 < b < 1
REM Ergebnis : s = artanh(b)

REM Methode: Simpsonsche Formel


DEF FN.f(x) = 1/(1-x*x)


PRINT "Bitte das Argument angeben, dessen Areatangens bestimmt"
INPUT "werden soll:   b = "; arg
WHILE ABS(arg) >= 1
    INPUT "Das Argument muß im Bereich (-1,1) liegen! b = "; arg
WEND
IF arg > 0 THEN a = 0: b = arg ELSE a = arg: b = 0

REM Berechnung des Maximums der vierten Ableitung
xm = ABS(arg)
xa = FN.f(xm)*FN.f(xm)*FN.f(xm)
xb = xa * FN.f(xm) * xm * xm
xc = xb * FN.f(xm) * xm * xm
m = 24*xa + 288*xb + 384*xc

INPUT "Bitte die Genauigkeit der Integration vorgeben: e = "; e
e = ABS(e)
lx = b-a
n = SQR( SQR(lx5*m/(2880*e)) )
IF n > 1 THEN n = INT(n) + 1 ELSE n = 2
h = lx/n

PRINT "Die vorgegebene Genauigkeit erfordert "; n;
PRINT "Schleifendurchläufe!"
PRINT "Bitte warten."

s = 0

FOR i = 1 TO n
    s = s + FN.f(a+(i-.5)*h)
NEXT
s = 2 * s

s = s + .5 * (FN.f(a)+FN.f(b))

FOR i = 1 to n-1
    s = s + FN.f(a+i*h)
NEXT

s = (s * h) / 3
IF arg < 0 THEN s = -s

PRINT "Berechneter Funktionswert: "; s
w = .5 * LOG( (1+arg) / (1-arg) )
PRINT "Wirklicher  Funktionswert: "; w
PRINT
PRINT "Differenz: "; ABS(s - w)

END
```

Programm 3. Eine geschickte Mischung aus Sehnentrapezen und Rechtecken ergibt die effiziente Simpson-Regel.

```
REM Programm zur Erstellung einer Tabelle der Integral - Sinus -
REM Werte im Bereich 0.1 ... 1
REM Der Integralsinus Si(x) ist definiert als Integral der
REM Funktion sin(t)/t über dem Intervall 0..x

REM Methode: Numerische Integration m. H. der Rechteckregel


DEF FN.f(x) = SIN(x)/x
REM Achtung bei x = 0 !


REM Bestimmung des Maximums der zweiten Ableitung
m = 2

PRINT "Bitte die Genauigkeit, mit der die Werte berechnet werden"
INPUT "sollen, angeben: e = "; e

e = ABS(e)

PRINT "x", "n", "Si(x)"
PRINT

FOR x = 0.1 TO 1.0 STEP 0.1
    n = INT(SQR(x*x*x*m/(24*e)))+1
    h = x/n
    PRINT x, n,
    s = 0
    FOR i = .5 TO n-.5
        s = s + FN.f(i*h)
    NEXT i
    s = s*h
    PRINT s
NEXT x

END
```

Programm 4. Der Integralsinus ist eine typische Aufgabe für die numerische Integration. ct



Erste Schritte zur analogen Eingabe

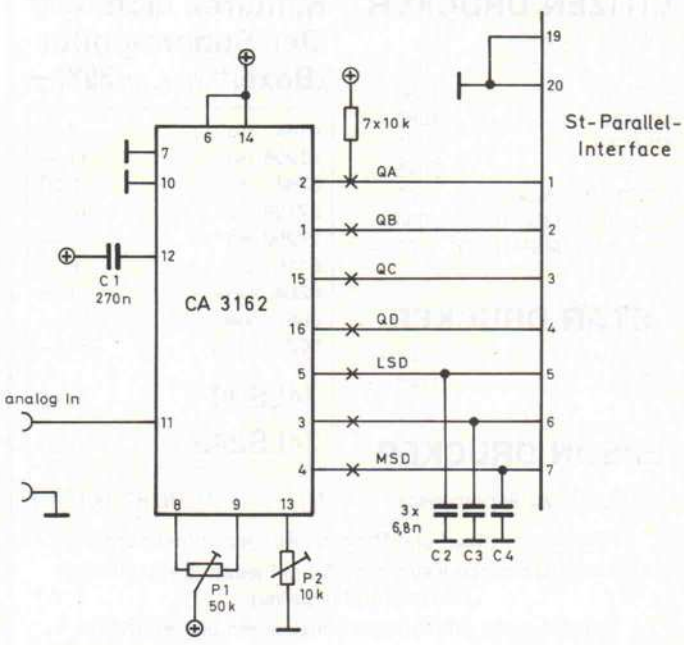
Einfacher AD-Wandler am c't-ST-Userport

Bernd Schöfer

Mit dem Anschluß eines einfachen AD-Wandlers erhält der Atari ST eine Schnittstelle zur 'analogen Umwelt'. Der Baustein CA 3162, dessen Betrieb an dem in c't 3/86 vorgestellten 'ST-Userport' hier beschrieben wird, eignet sich zur recht genauen Umsetzung von Spannungswerten in digitale Daten. Man kann mit dieser Erweiterung den ST zum Voltmeter umfunktionieren oder Meßwertaufnehmer wie etwa Temperaturfühler oder Drucksensoren anschließen, um physikalische Vorgänge in der Umwelt vom Computer erfassen und auswerten zu lassen.

Der CA 3162 ist leicht erhältlich und einigermaßen preisgünstig. Er liefert rund 100 Messungen pro Sekunde über einen Bereich von +999 mV bis -99 mV ab, besitzt eine Eingangsimpedanz von 100 MOhm und kommt mit nur einer Versorgungsspannung von +5 V aus. Die digitalen Daten liegen im BCD-Format (BCD: Binary Coded Decimal – binär kodierte Dezimalzahl) an vier Ausgangsleitungen an. Drei weitere Ausgänge zeigen an, welche der drei Dezimalstellen gerade ausgegeben wird.

Die Schaltung mit dem Wandler-IC haben wir auf einer kleinen Lochrasterplatine frei verdrahtet. Man benötigt lediglich sieben Widerstände (am besten ein Widerstandsarray), zwei Trimmer und drei Kondensatoren. Der Anschluß an den 'ST-Userport' über ein kurzes Stück Flachbandkabel dürfte keine Probleme bereiten. Allerdings muß man ein zusätzliches Kabel für die 5-V-Versorgung legen (am besten an R5 der Interface-Karte anlöten), da diese nicht auf die 50polige Steckerleiste geführt ist.



So wird der Wandler an das Parallel-Interface angeschlossen.

Die beiden Trimmpotis dienen zum Abgleich des Wandlers: Man stellt zunächst mit P1 bei kurzgeschlossenem Eingang den Nullpunkt ein. Dann legt man eine Spannung bekannter Höhe an und stellt P2 so ein, daß der Meßwert korrekt ausgegeben wird. Falls der negative Endwert von -99 mV nicht erreicht wird, kann C1 verkleinert werden. C2. . .C4 unterdrücken eventuelle Störungen.

Das nebenstehende Assembler-Programm für den Atari ST sorgt für die Anzeige des Meßwertes an einer festen Cursor-Position auf dem Bildschirm. Zuerst wird der Supervisor-Mode eingestellt, danach das Unterprogramm 'adw' aufgerufen. In diesem wird der Bildschirm gelöscht und eine Meldung ausgegeben. Das Programm nutzt bei allen Bildschirmausgaben intensiv die Escape-Funktionen, die GEM-DOS zur Verfügung stellt.

Im Programmteil 'adw0' holt der Rechner zyklisch die Hunderter-, Einer- und Zehnerstelle vom AD-Wandler, wandelt die Daten in das ASCII-Format um und stellt einen auszugebenden String zusammen. Dabei werden auch das Vorzeichen und ein eventueller Überlauf berücksichtigt. Der Wandler gibt beim Auftreten eines negativen Meßwertes eine '10' an der Hunderterstelle aus; ein Überschreiten des Meßbereichs in negativer Richtung wird durch Ausgeben einer '10' an jeder Stelle angezeigt, ein Überschreiten in positiver Richtung entsprechend durch eine '11'.

RATEV ELEKTRONIK-VERTRIEBS GMBH · 4030 RATINGEN 1 · Postfach 1601
Gothaerstr. 15 · Telefon: (02102) 42051 · (Mailbox 02102/475400) · Tx. 8585180

Rams

4164 150 nS	4,30
41256 150 nS	8,95
TMS 4416	11,50
6116 LP-3	3,50
6264 LP-15	7,30
6116 Flat Pack	11,90
6264 Flat Pack	25,90

Eproms

2532 450 nS	12,50
2716 450 nS	8,50
2732 250 nS	8,20
2764 250 nS	6,95
27128 250 nS	6,80
27256 250 nS	13,95

ICs für Commodore

6510	45,00
6526	39,90
6589	115,00
6581	78,00

Mikroprozessoren

Z80A CPU	4,40
Z80A PIO	5,30
Z80A CTC	5,30
Z80A DMA	14,20
Z80A DART	13,50
Z80A SIO/0	14,50
Z80A STI	39,90
Z80B CPU	7,10
Z80B PIO	10,50
Z80B CTC	10,50
Z80B DART	19,50
Z80B SIO/0	21,30
6502P	13,40
6502A	14,30
6504P	13,05
6504A	14,30
6520P	10,50
6520A	11,30
6522P	11,70
6522A	12,70
6532P	15,70
6532A	17,30
6551P	15,20
6551A	17,30
65C02 1 MHz	19,50
65C02 4 MHz	39,90
6800	9,85
6802	8,95
6809	22,90
6821	5,15
6840	14,50
6845	13,75
6850	4,90
68000 8 MHz	54,00
68008 8 MHz	72,50

Controller

UPD765	16,90
FD 1770	57,45
FD 1771	33,50
FD 1791	25,50
FD 1793	25,50
FD 1797	25,50
FD 2793	42,90
FD 2797	27,30
WD 1691	27,50
WD 2143	22,50
FDC 9216B	14,95
FDC 9229B	54,00

Grafik-ICs

EF 9365	70,00
EF 9366	66,00
EF 9367	75,00
UPD 7220	43,00

Sonder-ICs

UPD 70108 D8	34,90
UPD 70116 D8	39,90
HD 64180 6 MHz	119,00
AM 7910	79,00
AM 7911	79,00
AM 25 LS 2532	8,70
HD3-4702	24,90
MM 58167	48,75
MSM 5832	18,50
MSM 58321	18,50
RTC 58321	19,50
UPD 1990	9,40
8087 5 MHz	429,00

Quarzoszillatoren

alle gängigen Frequenzen ... 9,40

Kartenstecker (Floppy)

20 pol vergoldet	5,90
26 pol vergoldet	6,90
34 pol vergoldet	6,80
40 pol vergoldet	10,45
50 pol vergoldet	9,30

Quarze

0.032768	MHz	1,90
1.0	MHz	8,60
2.0	MHz	5,65
2.4576	MHz	5,50

alle anderen Standard-frequenzen ... 2,40

VG-Steckverbinder

64 Stift	2,25
64 Feder	2,90
96 Stift	4,85
96 Feder	5,25
64 Feder WW	4,85
96 Feder WW	7,00
64 Stift anschl.	28,90
64 Feder anschl.	15,90

D-Sub Steckverb. löt

Stift 9 pol	1,60
Stift 15 pol	1,80
Stift 25 pol	2,50
Stift 37 pol	3,90
Stift 50 pol	4,90
Feder 9 pol	2,05
Feder 15 pol	1,90
Feder 25 pol	3,40
Feder 37 pol	5,45
Feder 50 pol	7,90

D-Sub Steckverbinder Flachkabel

Stift 25 pol	8,70
Feder 25 pol	9,70

Gehäuse für D-Sub Steckverbinder

9 pol	2,35
15 pol	2,55
25 pol	2,65
37 pol	3,25
50 pol	3,35

Pfostenfederleisten 2-reihig

14 pol vergoldet	2,55
16 pol vergoldet	2,65
26 pol vergoldet	3,60
34 pol vergoldet	3,85
40 pol vergoldet	5,55
50 pol vergoldet	6,95

Pfostenstiftleisten mit Verpolungsschutz gerade

14 pol vergoldet	1,65
16 pol vergoldet	1,85
26 pol vergoldet	2,70
34 pol vergoldet	3,70
40 pol vergoldet	4,25
50 pol vergoldet	5,10

Pfostenstiftleisten 90°

14 pol vergoldet	2,95
16 pol vergoldet	3,05
26 pol vergoldet	4,15
34 pol vergoldet	5,45
40 pol vergoldet	5,95
50 pol vergoldet	7,55

Dilstecker für Flachkabel

14 pol	1,55
16 pol	1,70
24 pol	2,15
28 pol	2,35
40 pol	2,95

Centronics-Stecker

24 pol lötbar	9,20
36 pol lötbar	6,90
50 pol lötbar	15,80
36 pol Flachkabel	8,50

Centronics-Buchsen

24 pol lötbar	9,40
36 pol lötbar	7,90
50 pol lötbar	16,45
36 pol Flachkabel	9,80

Textool-Fassungen

16 pol	20,90
24 pol	21,30
28 pol	22,50
40 pol	31,30

Burn-In Fassungen

16 pol	7,85
24 pol	11,90
40 pol	13,85

Vielschicht Kondensatoren 0,1 µF Raster 2,5 oder 5 mm

10 Stück	à 0,28
100 Stück	à 0,24
500 Stück	à 0,21

Low-cost IC Sockel

8—40 pol p. Pin 0,02

Präzisions IC Sockel

8—40 pol p. Pin	0,05
64 pol	4,50

Flachbandleitung grau pro Ader u. Meter 0,10

Weitere Steckverbinder auf Anfrage

IBM-Printerkabel 31,00

Liefer- und Zahlungsbedingungen: Die genannten Preise verstehen sich inkl. MwSt. Die Lieferung erfolgt per Nachnahme + Porto und Versandspesen. Die Angebote sind freibleibend. Zwischenverkauf vorbehalten. Der Mindestbestellwert beträgt DM 30.00.

CITIZEN DRUCKER

2 Jahre Garantie

120 D	880,—
MSP 10	1680,—
MSP 15	1983,—
MSP 20	2295,—
MSP 25	2945,—
RS 232 Interface	125,—
Einzelbl. f. MSP 10/20	690,—
Einzelbl. f. MSP 15/25	780,—

STAR DRUCKER

SG 10	858,—
SG 15	1189,—

EPSON DRUCKER

ab Lager zu Superpreisen

Hantarex Monitore
Der Supermonitor
„Boxer“ WS o. SW **398,—**

4164	150	4,—
41256	150	11,—
2764	250	6,50
27128	250	10,—
27256	250	18,—
6116	LP3	6,—
6264	PP15	11,—
8087	8 MHz	555,—
80287	6 MHz	655,—
74LS00		0,60
74LS245		2,30

ab Lager 74..LS, N, HC, HCT, S, F

Zwischenverkauf und Preisänderungen vorbehalten.

Wir konfektionieren Kabel aller Art auf jeden Steckverbinder schnell und preiswert.

Wir entwerfen und fertigen Schaltungen und Konzepte (Klein- bis Mittelserien) im Computer- und NF-Bereich.

Großes Angebot aus der gesamten Elektronik zu guten Preisen.
Wir liefern nur 1. Wahl Industriequalität.

BS ELEKTRONIK

Tel. 0 69-4 98 03 33, Sandweg 38, 6000 Ffm 1


```
rinpin  equ $fafff9                     adr.des low-inputport
hund    equ 6                           msb-stelle des 3162
zehn    equ 5                            "
ein     equ 4                           lsb   "
gem     equ 1                           dos
print   equ 9                            " funktion:ausgabe string
testinp equ 11                           "    "     :test keyboard
dircon  equ 7                            "    "     :hole char.


        .text

start   clr.l -(sp)                     init superv.state
        move.w #$20,-(sp)
        trap #gem
        add.l #6,sp
        move.l d0,spsave                save superv.stack
        bsr adw                         beispielprog. ausfuehren
        move.l spsave,-(sp)             init user mode
        move.w #$20,-(sp)
        trap #gem
        add.l #6,sp
        clr.w -(sp)                     ab nach GEM
        trap #gem


*****   adw.ausgabe am bildschirm bis ESC

adw     lea clear,a2                    clear screen
        bsr pritxt
adw0    move.b rinpin,d2                hole adw-data 100
        btst #hund,d2
        bne adw0
        andi.b #$f,d2
        cmpi.b #10,d2                   ueberlauf ? u.test (+/-)
        bhi admist
        bne adplus
        move.b #'-',sign                vorzeichen in string basteln
        clr.b d2
        bra adw01
adplus  move.b #'+',sign
adw01   addi.b #'0',d2                  umwandeln in ascii
        move.b d2,data100
adw1    move.b rinpin,d2                hole adw-data 1
        btst #ein,d2
        bne adw1
        andi.b #$f,d2
        cmpi.b #9,d2                    >9 = pos/neg ueberlauf
        bhi admist
        addi.b #'0',d2
        move.b d2,data1
adw2    move.b rinpin,d2                hole adw-data 10
        btst #zehn,d2
        bne adw2
        andi.b #$f,d2
        cmpi.b #9,d2
        bhi admist
        addi.b #'0',d2
        move.b d2,data10
        lea ptext,a2
        bsr pritxt
        bsr tstkey
        bcc adw0
bye     rts

tstkey  move.w #testinp,-(sp)           test keyboard
        trap #gem
        addq.l #2,sp
        tst.w d0
        bne tstke
tstk    andi.b #$fe,ccr                 clc
        rts
tstke   move.w #dircon,-(sp)
        trap #gem
        addq.l #2,sp
        cmpi.b #$1b,d0                  ESC ?
        bne tstk
        ori.b #1,ccr                    sec
        rts

pritxt  move.l a2,-(sp)
        move.w #print,-(sp)
        trap #gem
        addq.l #6,sp
        move.w #1,ovflag
        rts

admist  tst.w ovflag
        beq admis
        lea over,a2
        bsr pritxt
        clr.w ovflag
admis   bsr tstkey
        bcc adw0
        bra bye


        .data
        .even

spsave    dc.l 0                        stackbuffer
ovflag    dc.w 1
clear     dc.b 27,'E',27,'Y',34,45,'**** c't mini-voltmeter ****',27,'f',0
over      dc.b 7,27,'Y',45,45,27,'K',27,'p','overflow !!',27,'q',7,0
ptext     dc.b 27,'Y',45,45
sign      dc.b '+'
data100   dc.b '0'
data10    dc.b '0'
data1     dc.b '0 mV      ',0

        .end
```

Treiberprogramm für den AD-Wandler CA 3162 am 'ST-Userport'

Bitverdoppler

Ein Bit Speicher – zwei bit Information

Bernd Wiegmann

Speicherplatz war noch vor kurzer Zeit so teuer, daß man mit jedem Byte geizen mußte. Doch sind in jüngster Zeit die Preise derart gefallen, daß Speichergrößen zum Standard wurden, von denen man vor ein paar Jahren noch nicht einmal träumte. Jetzt scheint ein weiterer großer Schritt in diese Richtung bevorzustehen, nachdem in den Labors eines amerikanischen Herstellers eine Möglichkeit entwickelt worden ist, Speicherplatz doppelt zu nutzen.

Die wundersame Speicherplatzvermehrung wird durch Wandler-Elemente bewirkt, die einfach in die Eingangs- und Ausgangsleitungen von herkömmlichen dynamischen RAMs geschaltet werden. Beim näheren Studium der Datenblätter, die zur Zeit noch den Stempel 'Preliminary' tragen, erkennt man bald die simple Idee hinter diesem Geniestreich: Am besten kann man die Funktion verstehen, wenn man sich den Speicher als einen großen Schrank mit vielen Schubfächern vorstellt, von denen jedes ein Bit zu speichern vermag. Ist ein Fach gefüllt, repräsentiert es den Wert 1, andernfalls den Wert 0.

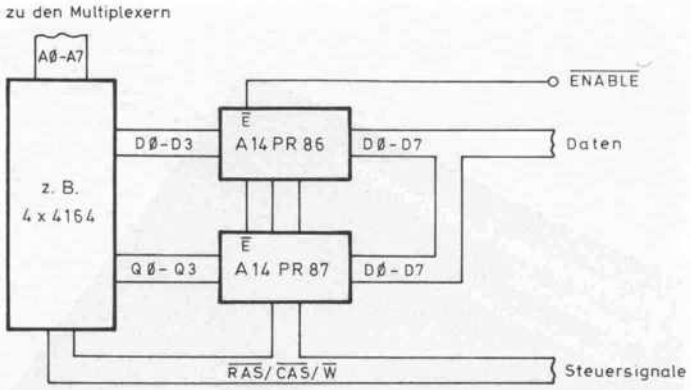
Was aber nun, wenn das Fach nur zur Hälfte gefüllt ist? Bei statischen Speichern kann es durch den rein digitalen Aufbau keinen Zwischenstand geben. Anders bei dynamischen Speichern. Diese Bauelemente enthalten bekanntlich Kondensatoren, und bei Kondensatoren kann man sehr wohl verschiedene Spannungen detektieren. Rein theoretisch ist eine beliebig feine Abstufung möglich. Freilich arbeitet man mit sehr kleinen Ladungsmengen, weshalb die jetzt vorgestellten 'DRAM Doubler' nur vier Zustände unterscheiden können. Im Laborexperiment soll jedoch unter Anwendung einer speziellen Refresh-Technik sogar schon eine 8-Bit-Auflösung gelungen sein.

Fachleute werden erkennen, daß es sich bei den neuen Bauelementen im Prinzip um schlichte D/A- und A/D-Konverter handelt. Der Chip mit der Typenbezeichnung A14 PR 86IL wandelt das 2-Bit-Eingangsdatum in vier Spannungspegel um, der A14 PR 87IL bildet beim Auslesen der Spannungspegel daraus wieder eine 2-Bit-Information.

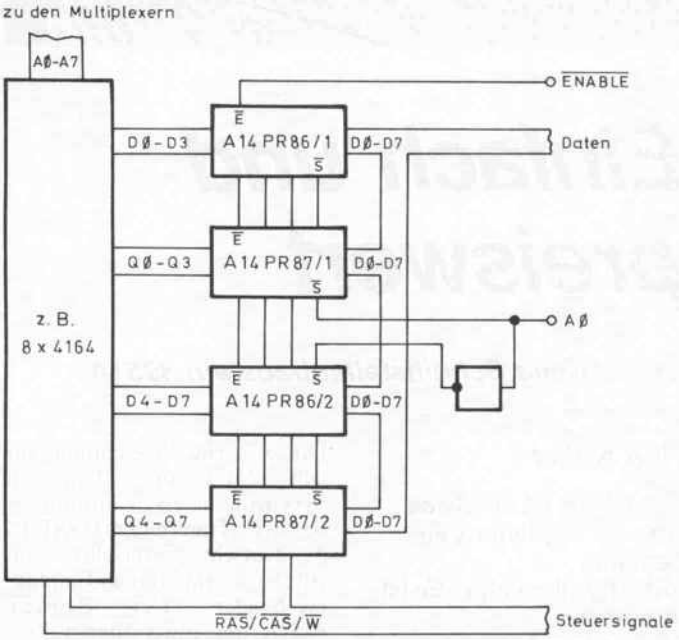
Schaltungspraxis

Besonders interessant wird die Neuentwicklung dadurch, daß je nach Anwendung entweder die Speicherkapazität verdoppelt oder die Zugriffszeit halbiert werden kann. Damit wird die Möglichkeit eröffnet, mit herkömmlichen dynamischen RAMs Zugriffszeiten in einer Größenordnung von 60 bis 75ns zu erzielen – sehr wichtig für Rechnerkonzepte, in denen die neuen RISC-Prozessoren (RISC: Reduced Instruction Set Computer) mit extrem knappen Zykluszeiten zum Einsatz kommen.

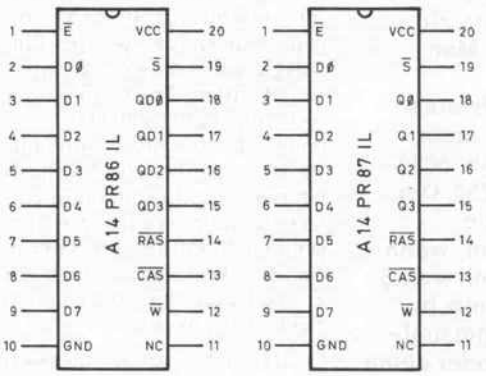
Die nebenstehenden Skizzen verdeutlichen die alternativen Schaltungskonzepte: Im ersten Fall wird die Anzahl der Speicherelemente bei gleichbleibender Kapazität halbiert, doch ansonsten ist die gesamte Schaltung konventionell aufgebaut. Im zweiten Fall wird die Zugriffsgeschwindigkeit verdoppelt. Man erkennt, daß bei dieser Version die niedrigstwertige Adresse (A0) durch eine Zusatzschaltung erzeugt werden muß. Es besteht die Einschränkung, daß immer zwei aufeinanderfolgende Adressen nacheinander angesprochen werden müssen (was bei 16-Bittern ohnehin der Fall ist). Die restliche Schaltung ist herkömmlich aufzubauen, nur können eventuell Datenbus-Treiber entfallen, da die neuen Wandler deren Funktion ebenfalls übernehmen.



Schaltung zur Kapazitätsverdopplung



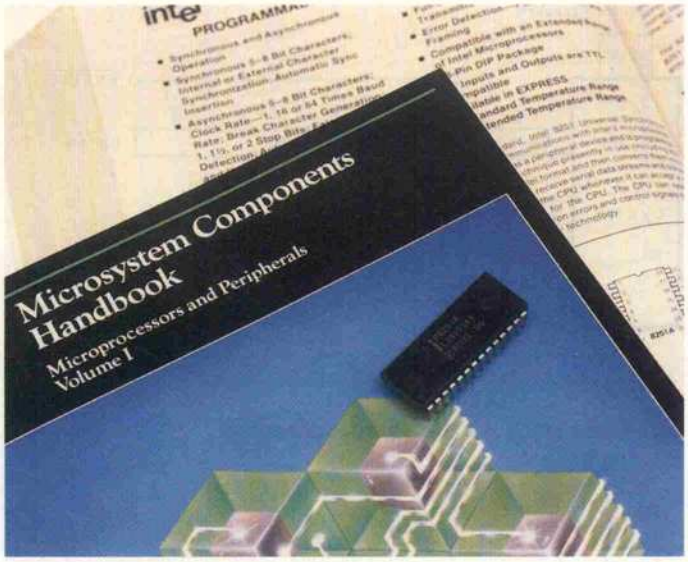
Schaltung zur Geschwindigkeitsverdopplung



Anschlußbelegung der neuen Wandler-ICs

Literatur

Datenblätter der Bausteine A14 PR 86IL und A14 PR 87IL sind zu beziehen bei: General Dynamics Improvisation, Box 141986, Silentio Valley, USA.



Einfach und preiswert

Der serielle Schnittstellenbaustein 8251A

Rolf Keller

Der 8251A ist so etwas wie der VW unter den seriellen Schnittstellenchips. Er ist in einem platzsparenden 28-Pin-Gehäuse untergebracht, seine Innereien sind recht unkompliziert, und er ist auch recht einfach zu programmieren. Man kann dieses IC zusammen mit diversen CPUs betreiben, unter anderem mit 8048, 8080, 8085, 8086 oder Z80. Der 8251A ist nicht die schlechteste Wahl, wenn es darum geht, mit wenig Aufwand eine einfache asynchrone Schnittstelle für ein Terminal oder einen Drucker aufzubauen. Als Terminalschnittstelle wird dieser Chip beispielsweise im c't86, unserem 16-Bit-Selbstbaurechner mit 8086-CPU, verwendet.

Die offizielle Bezeichnung für den 8251A lautet 'Universal Synchronous/Asynchronous Receiver/Transmitter (USART), zu deutsch: universeller, synchroner/asynchroner Empfänger/Sender. Der Baustein 8251A hat einen älteren Vorgänger, den 8251 (ohne A). Die Neuauflage war erforderlich, weil der alte 8251 einige Mängel und 'unpraktische Eigenarten' hat, die manche Anwendungen stark erschweren. Beide Versionen besitzen die gleiche Pin-Belegung und die gleichen internen Register. Trotzdem ist der A-Typ in bezug auf Hardware und Software nicht voll aufwärtskompatibel zu seinem älteren Bruder. Es sind dafür einfach zu viele Änderungen vorgenommen worden, nicht nur in der Logik, sondern auch im Zeitverhalten und sogar in den elektrischen Werten.

Die Hersteller liefern nur noch den A-Typ, aber in dem einen oder anderen Lager eines Händlers findet sich vielleicht noch die alte Version. Deren Einsatz sollte man aber möglichst vermeiden. Das Intel-Datenbuch nennt elf Verbesserungen, aus denen man unter anderem ersehen kann, daß der 8251 den Sendevorgang mitten im Transfer eines Byte abbrechen kann, was beim 8251A unterbunden wurde.

Der 8251A bietet je einen seriellen Empfangs- und einen Sendekanal, die asynchron (Baudrate null bis 19200) oder synchron (Baudrate null bis 64000) arbeiten können. Allerdings sind beide Kanäle nicht vollkommen unabhängig voneinander programmierbar.

Beschaltung

Bild 1 zeigt die Pin-Belegung des Bausteins. Alle Signal-Anschlüsse sind TTL-kompatibel. An den Rechnerbus kann der Chip in üblicher Weise angeschlossen werden. Die vier adressierbaren Register werden von der CPU (bei $\overline{CS} = 0$) wie folgt angesprochen:

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
0	0	1	Empfangsdaten-register
0	1	0	Sendedaten-register
1	0	1	Statusregister
1	1	0	Befehlsregister

Dem Pin CLK muß ein externer Takt (0,741 bis 3,125 MHz) zugeführt werden. Keines der anderen Signale ist fest mit CLK gekoppelt; der Takt dient lediglich dazu, den Chip 'in Gang zu halten' (dynamische Register).

Der Bezugstakt für die Baudrate wird an $\overline{RxC}$ (für den Empfänger, Receiver Clock) und – eventuell mit anderer Frequenz – an $\overline{TxC}$ (für den Sender, Transmitter Clock) zugeführt. Die niedrigste zulässige Frequenz ist null, die höchstzulässige ist für den konkreten Anwendungsfall durch eine Art Puzzlespiel zu ermitteln.

Erstens können nur Baudraten von maximal 19200 (asynchron) beziehungsweise 64000 (synchron) benutzt werden. Zweitens gelten folgende Höchstwerte: 64 kHz (bei einem programmierten Baudratenfaktor '× 1'), 310 kHz ('× 16') und 615 kHz ('× 64'). Drittens darf die Frequenz von $\overline{RxC}$ und $\overline{TxC}$ nicht höher sein als 1/30 der CLK-Frequenz (Pin 20) beim Baudratenfaktor '× 1' beziehungsweise 1/4,5 bei '× 16' und '× 64'. Sende- und Empfangsdaten laufen seriell über die Pins TxD (Transmit Data) und RxD (Receive Data).

Die Ausgänge RxRDY, TxRDY und TxEMPTY können sowohl als Interrupt-Ausgänge als auch als DMA-Steuersignale benutzt oder einfach offengelassen werden. Nur mittels dieser Signale ist Interrupt-Betrieb möglich. Vektor-Interrupts kennt der 8251A nicht, weshalb man in der Regel dafür einen Interrupt-Controller-Baustein zusätzlich einsetzen wird.

Der Pin 16 arbeitet im Asynchron-Modus als BRKDET-Ausgang (Break Detect, Unterbrechungserkennung). Er zeigt an, daß das Empfangsdaten-Signal RxD für die Zeitdauer mindestens eines Daten-Byte auf logisch '0' (entspricht einer unterbrochenen Leitung) liegt. Diese Funktion wird allerdings selten ausgenutzt.

Im Synchron-Modus arbeitet Pin 16 als SYNDET (Synchronisation Detect, Synchronisationserkennung), wahlweise als Eingang oder Ausgang programmierbar. Bei Verwendung als Eingang ist eine externe Zusatzschaltung erforderlich, die den Zeitpunkt erkennt, in dem auf der Empfangsdatenleitung RxD der Anfang eines Daten-Byte 'vorbeiläuft'. Diese Schaltung gibt dann einen SYNDET-Impuls an den 8251A, dieser rastet daraufhin mit seiner Empfangsschaltung auf den Byte-Anfang ein.

Ist SYNDET als Ausgang programmiert, so sucht der 8251A selbst den Empfangsdatenstrom nach den programmierten SYNC-Zeichen (ein oder zwei) ab und meldet über den SYNDET-Ausgang, wenn seine Empfangsschaltung darauf eingerastet ist.

Die Ausgänge $\overline{RTS}$ und $\overline{DTR}$ können per Programm auf logisch '0' oder '1' gesetzt, der Eingang $\overline{DSR}$ kann per Programm abgefragt werden. Es gibt intern keinen hardwaremäßigen Zusammenhang dieser drei Signale untereinander oder mit sonstigen Funktionen des Bausteins. Sie sind frei für beliebige Verwendung, zum Beispiel zur Verbindung mit den gleichnamigen Signalen eines Modems, müssen aber per Software bedient werden.

Anders ist es beim Eingang $\overline{CTS}$. Legt man diesen auf logisch '1', so wird der Senderausgang TxD bedingungslos gesperrt. Damit ergibt sich eine simple Handshake-Möglichkeit, unabhängig vom Programm.

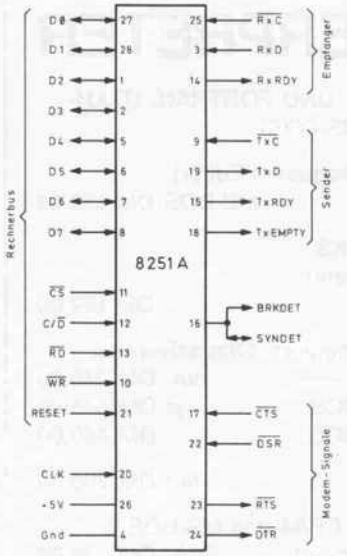


Bild 1. Pinbelegung des 8251(A)

Programmierung

Die Programmierung des Bausteins ist ziemlich einfach. Man unterscheidet Modus-Bytes und Kommando-Bytes (Tabelle 1). Beide werden in das einzige adressierbare Befehlsregister geschrieben. Zu Anfang schreibt man stets ein Modus-Byte, mit dem die grundsätzliche Betriebsart (Asynchron- oder Synchron-Modus) eingestellt wird. Falls man 'Synchron-Modus' gewählt hat, müssen unmittelbar danach ein beziehungsweise zwei Bytes folgen, die später als SYNC-Zeichen-Muster dienen.

Nach dem Modus-Byte sind Sender und Empfänger auf jeden Fall gesperrt. Mit einem darauffolgenden Kommando-Byte kann man sie unabhängig voneinander freigeben, wodurch der Baustein zur Datenübertragung bereit wird. Während der Übertragung kann beliebig oft ein weiteres Kommando-Byte folgen; das ursprüngliche Modus-Byte bleibt bis zu einem Reset – per Hardware oder Software – gültig. Mit diesen weiteren Kommando-Bytes steuert man zum Beispiel die Übertragungsrichtung bei Halbduplex, löscht die Fehler-Bits im Statusregister und schaltet die Ausgänge $\overline{RTS}$ und $\overline{DTR}$.

Bei der Initialisierung geht es etwas 'hakelig' zu. Nach den einzelnen Bytes vom Modus-Byte bis zum ersten Kommando-Byte muß das Programm eine Wartezeit von mindestens 6 CLK-Zyklen einlegen. Dies ist nötig, weil zunächst kein Status-Bit anzeigt, wann geschrieben werden darf. Nach der Initialisierung gibt das TxRDY-Bit (Bit 0 im Statusregister) die Erlaubnis, Kommando-Bytes oder Sendedaten in den Baustein zu schreiben. Ein Kommando-Byte, bei dem Bit 6 (Reset) gesetzt ist, bewirkt den Abbruch aller Funktionen; anschließend erwartet der Baustein wieder ein Modus-Byte.

Hat man kein Hardware-Reset-Signal zur Verfügung, so muß man nach dem Einschalten einen Reset programmieren, am besten mit dem Kommando-Byte $40. Weil der Baustein aber zunächst in einem undefinierten Zustand ist, weiß man nicht, ob er als erstes ein Modus-Byte, ein SYNC-Byte oder ein Kommando-Byte erwartet. Eine Folge von Befehls-Bytes, die immer funktioniert, ist beispielsweise $00, $00, $00, $40. Nach den drei Null-Bytes erwartet der 8251A mit Sicherheit ein Kommando-Byte. Aber Achtung: die Ausgänge $\overline{RTS}$ und $\overline{DTR}$ werden dabei – vielleicht ungewollt – auf logisch '1' gesetzt.

Das einzige adressierbare Statusregister des Bausteins kann per Programm jederzeit gelesen werden. Allerdings ist es nicht immer auf dem neuesten Stand. Aktuelle Änderungen sind im Statusregister erst nach einer Verzögerungszeit von bis zu 28 Taktzyklen (CLK-Pin 20) erkennbar. Achtung: beim 8251 sind es nur 16 Zyklen, einige 8251-Programme laufen deswegen nicht mit der A-Version.

Für den asynchronen Sendebetrieb ist Bit 4 des Modus-Byte ohne Bedeutung. Der programmierte Baudratenfaktor (Bit 0,1) gibt an, um wieviel der Bezugstakt $\overline{TxC}$ höher (1fach, 16fach, 64fach) als die tatsächliche Baudrate sein muß. Der Sender (Ausgang TxD) ist nur dann freigegeben, wenn sowohl Bit 0 im Kommando-Byte gesetzt ist, als auch der $\overline{CTS}$-Eingang auf logisch '0' liegt.

Asynchron senden . . .

Bei nicht freigegebenem Sender liegt TxD auf logisch '1', es sei denn, es ist das Bit 3 im Kommando-Byte gesetzt. Bei freigegebenem Sender kann man per Programm Daten senden, gesteuert entweder durch Polling des Statusregisters oder – bei geeigneter Schaltung – durch Interrupts und/oder DMA mit Hilfe der Signale TxRDY und TxEMPTY.

Beim Polling des Statusregisters ist der feine Unterschied zwischen TxRDY (Bit 0) und TxEMPTY (Bit 2) zu beachten, siehe Tabelle 2. Im Normalfall wird man TxRDY benutzen. Damit funktioniert ohne weitere Maßnahmen das Handshaking über den $\overline{CTS}$-Pin. Außerdem kommt dann der eingebaute Puffer des Senders zur Wirkung: TxRDY ist stets gesetzt, wenn ein Platz im Puffer frei ist, unabhängig davon, ob der Pufferinhalt schon gesendet ist.

TxEMPTY benutzt man nur dann, wenn man sich per Programm vergewissern möchte, daß der Puffer leer ist, also alle Daten schon beim entfernten Empfänger angekommen sind. Diese Information wird aber selten benötigt, weil die Konstruktion des 8251A (im Gegensatz zum 8251) garantiert, daß die Sperre des Senders (per Programm oder per $\overline{CTS}$-Signal) immer so lange aufgeschoben wird, bis der Puffer leer ist.

. . . und empfangen

Für den asynchronen Empfangsbetrieb sind die Bits 6 und 7 des Modus-Byte ohne Bedeutung: der Empfänger prüft stets auf ein Stop-Bit; falls noch weitere folgen, stört ihn das nicht. Der Baudratenfaktor '× 1' wird selten gewählt, weil dabei eine Zusatzschaltung erforderlich ist, die den Takt $\overline{RxC}$ nicht nur mit der richtigen Frequenz (wie beim Sender erforderlich), sondern auch mit der richtigen Phasenlage liefert.

Modus-Byte für Synchron-Modus:	
Bit 7	0: zwei SYNC-Zeichen 1: ein SYNC-Zeichen
Bit 6	0: SYNDET-Pin ist Ausgang 1: SYNDET-Pin ist Eingang
Bit 5	0: ungerade Parität 1: gerade Parität
Bit 4:	0: ohne Paritätsprüfung 1: mit Paritätsprüfung
Bit 3,2	00: Zeichenlänge 5 Bit 01: Zeichenlänge 6 Bit 10: Zeichenlänge 7 Bit 11: Zeichenlänge 8 Bit
Bit 1,0	stets: 00
Modus-Byte für Asynchron-Modus:	
Bit 7,6	00: unzulässig 01: 1 Stop-Bit 10: 1½ Stop-Bits 11: 2 Stop-Bits
Bit 5	0: ungerade Parität 1: gerade Parität
Bit 4	0: ohne Paritätsprüfung 1: mit Paritätsprüfung
Bit 3,2	00: Zeichenlänge 5 Bit 01: Zeichenlänge 6 Bit 10: Zeichenlänge 7 Bit 11: Zeichenlänge 8 Bit
Bit 1,0	00: unzulässig (bedeutet Synchron-Modus) 01: Baudratenfaktor x1 10: Baudratenfaktor x16 11: Baudratenfaktor x64
Kommando-Byte:	
Bit 7	0: ohne Wirkung 1: Suche nach SYNC-Zeichen beginnen (nur im Synchron-Modus)
Bit 6	0: ohne Wirkung 1: Reset des Bausteins (anschließend muß ein Modus-Byte folgen)
Bit 5	0: $\overline{RTS}$-Ausgang auf log. 1 setzen 1: RTS-Ausgang auf log. 0 setzen
Bit 4	0: ohne Wirkung 1: Fehler-Bits im Statusregister löschen
Bit 3	0: ohne Wirkung 1: Leitungsunterbrechung simulieren (TxD konstant auf log. 0 setzen)
Bit 2	0: Empfänger RxD sperren 1: Empfänger RxD freigeben
Bit 1	0: $\overline{DTR}$-Ausgang auf log. 1 setzen 1: $\overline{DTR}$-Ausgang auf log. 0 setzen
Bit 0	0: Sender TxD sperren 1: Sender TxD freigeben

Tabelle 1. Befehls-Bytes

Bit 7	Zeigt den invertierten Zustand des $\overline{\text{DSR}}$-Eingangs an.
Bit 6	Zeigt den Zustand des SYNDET/BRKDET-Pins an.
Bit 5	Fehler-Bit 'Frame error' Das Bit wird gesetzt, wenn der Empfänger im Asynchron-Modus das Fehlen eines Stop-Bits erkannt hat.
Bit 4	Fehler-Bit 'Overrun' Das Bit wird gesetzt, wenn die CPU dem Empfänger ein oder mehrere Daten-Bytes nicht rechtzeitig durch Lesen des Datenregisters abgenommen hat, so daß der Puffer des Empfängers übergelaufen ist (bedeutet Datenverlust).
Bit 3	Fehler-Bit 'Parity error' Das Bit wird gesetzt, wenn der Empfänger ein Daten-Byte mit Paritätsfehler empfangen hat.
Bit 2	TxEMPTY Identisch mit dem Zustand des gleichnamigen Pins. Das Bit ist gesetzt, wenn der Sender zwar freigegeben ist (Bit 0 im Kommando-Byte), aber keine Daten-Bytes zum Aussenden von der CPU her zur Verfügung hat: entweder sind alle Daten-Bytes schon ausgesendet, oder aber die CPU hat noch keine in das Datenregister geschrieben.
Bit 1	RxRDY Identisch mit dem Zustand des gleichnamigen Pins. Das Bit ist gesetzt, wenn der Empfänger freigegeben ist (Bit 2 im Kommando-Byte) und ein empfangenes Daten-Byte zum Lesen durch die CPU im Datenregister bereitsteht.
Bit 0	TxRDY Das Bit ist (unabhängig von der Freigabe des Senders) gesetzt, wenn die CPU ein auszusendendes Daten-Byte in das Datenregister oder in das Befehlsregister schreiben darf. Der Ausgangs-Pin TxRDY liegt auf log. 1, wenn das TxRDY-Bit gesetzt ist und zusätzlich der Sender durch Bit 0 im Kommando-Byte und durch $\overline{\text{CTS}}$ = log. 0 freigegeben ist.

Tabelle 2. Status-Register

Der Empfänger (Eingang RxD) ist nur dann freigegeben, wenn Bit 2 im Kommando-Byte gesetzt ist. Setzt man das Bit zu einem Zeitpunkt, bei dem schon Daten am Eingang liegen, so kann das erste empfangene Daten-Byte dadurch verfälscht werden, daß es 'angeschnitten' wird.

Bei freigegebenem Empfänger kann man per Programm Daten empfangen. Ähnlich wie beim Sender ist Polling des Statusregisters (Bit 1, RxRDY) oder Benutzung von Interrupts und/oder DMA (über RxRDY-Pin) möglich.

Ob ein empfangenes, aus dem Datenregister des 8251A gelesenes Byte fehlerfrei ist, erkennt man an den Fehler-Bits (Bit 3, 4 und 5) des Statusregisters. Wenn der Baustein einen Fehler erkennt, so setzt er zwar das entsprechende Fehler-Bit, arbeitet aber im übrigen völlig normal weiter. Komfortable Software kann man also mit einer Fehlerbearbeitung versehen. Die Fehler-Bits bleiben nach Auftreten eines Fehlers so lange stehen, bis sie per Programm durch Ausgabe eines Kommando-Byte mit gesetztem Bit 4 gelöscht werden. Leider ist von der CPU her nicht absolut sicher zu erkennen, zu welchem Daten-Byte ein Fehler-Bit gehört. Im Puffer des Empfängers kann sich ja mehr als ein Daten-Byte befinden, die Fehler-Bits existieren aber nur einmal.

Synchron?

Der Synchronbetrieb hat für den 8251A heutzutage keine große Bedeutung mehr. Es gibt inzwischen wesentlich leistungsfähigere Bausteine. Um ein modernes, synchrones Übertragungsprotokoll wie etwa HDLC (bei Datex-P benutzt) abzuarbeiten, benötigt der 8251A ein erhebliches Maß an Softwareunterstützung, da ihm viele Hardwarefunktionen einfach fehlen. Es ist ausgesprochen kompliziert, eine solche Software zu entwickeln. Außerdem wird man sich dann vermutlich auf sehr niedrige Baudraten beschränken müssen. Deshalb wird die synchrone Betriebsweise hier nicht weiter beschrieben. ct

COMPILER — INTERPRETER

★ NEVADA/UTAH COBOL- UND FORTRAN, UTAH-BASIC-Interpreter (nur MS-DOS)

★ NEVADA/UTAH EDIT (Programm-Editor)
CP/M-80 DM 129,00 MS-DOS DM 159,00

★ SOFTWARE TOOLWORKS
C-Compiler, LISP-Interpreter
CP/M-80 und MS-DOS DM 189,00

★ TURBO-PASCAL 3.0, amerikan. Originalversion
MS-DOS/PC-DOS: ——> nur: DM 210,00
8087 und BCD MS/PC-DOS je DM 445,00
8087 — BCD MS/PC-DOS DM 460,00

★ SIDEKICK engl. nur DM 205,00

★ USERGROUP Disketten CP/M und MS-DOS
(Inhaltsverzeichnis anfordern) Stck. DM 28,00

Weitere PC-Software und Hardware-Erweiterungen, auch für die NCR-DM V im Angebot.

Wir sind Vertragspartner für TANDON-COMPUTER.

Preise
incl. MwSt.

IST INSTITUT FÜR SOFTWARETECHNOLOGIE
Ekherstraße 19 · D-8904 Friedberg
Telefon (08 21) 60 43 44

C COMPILER

MI-C für CP/M, CP/M 86, MS-DOS

vereint hohen Bedienungskomfort mit hervorragender Leistung

- Vollständige Version mit 13stelliger BCD-Arithmetik für Gleitkommazahlen
- Erzeugt kurze und schnelle Programme, die auch in ein ROM gebracht werden können.
- Ausgabe in Z80-, 8080-, 8086-Assemblercode
- Kompatibel zu M80/L80 (MASM) von Microsoft
- Fehlerverfolgung mittels Trace möglich
- Umfangreiche Bibliothek incl. math. Funktionen
- für MS-DOS/CP/M 86: 4 Speichermodelle
- 8087 Math. Prozessor Unterstützung enthalten
- AMD 9511 Unterstützung erhältlich
- Unix-kompatibel
- Deutsche oder englische Version lieferbar
- 8"-/5,25"-/3,5"-/3"-Disk + deutsches Handbuch

MI-C für CP/M ..... **445,— DM**
MI-C für CP/M 86, MS-DOS ..... **575,— DM**
MI-C Crosscompiler ..... **745,— DM**
MI-C Crossassembler + Linker ..... **545,— DM**
MI-C AMD 9511 Unterstützung ..... **798,— DM**

Herbert Rose, Bogenstraße 32, 4390 Gladbeck, Telefon (0 20 43) 2 49 12 oder 4 35 97

Vertrieb in Österreich:
Dr. Willibald Kraml, Microcomputer-Software, Degengasse 27/16, A-1160 Wien

Härtetest

Die c't68ECB-Hardware in der Praxis

Peter Sager
Manfred Bertuch

Erste Erfahrungen liegen inzwischen vor. Für die neue 68000-CPU-Karte für das c't86-System schlug die Stunde der Wahrheit. Bei den 68000-Pionieren unter den Lesern und im c't-Labor mußte sie zeigen, was in ihr steckt. Nach der Beseitigung einiger Startschwierigkeiten erwies sich die neue CPU-Karte als willige und gutmütige Zugmaschine, die der c't86-Peripherie neuen Schwung verleiht. Im c't-Labor absolvierte sie nicht nur ihre Pflichtläufe, sondern glänzte auch in der Kür mit satten Reserven.

Die Aufgabenstellung war schnell umrissen. Für das existierende und mittlerweile verbreitete 16-Bit-Selbstbau-System auf der Basis der 8086-CPU sollte eine neue CPU-Karte mit dem leistungsstarken Motorola-Prozessor her, die in der unveränderten Hardware-Umgebung des c't86 läuft. Dies bedeutete auch eine Anpassung des asynchronen 68000-Timings an den synchronen c't86-Bus, der auf dem verbreiteten ECB-Standard aufbaut. Während der 68000 von Haus aus mit unterschiedlich schnellen Peripherie-Karten zusammenarbeiten kann und seine Schreib- und Lese-Zyklen so lange ausdehnt, bis die Peripherie mit dem Data-Acknowledge-Signal (DTACK) ihr o.k. gibt, müssen sich in synchronen Systemen alle Karten nach einem starren Zeitraster richten. Ein DTACK-Signal ist im ECB-Bus nicht vorgesehen, für den 68000 muß deshalb eins auf der CPU-Karte erzeugt werden. Wie dies bewerkstelligt wird, ist nicht ganz ohne Bedeutung für das Zusammenspiel mit der RAM-Karte, wie später noch gezeigt wird.

Ein weiteres Merkmal der neuen CPU-Karte sind bis zu 128 KByte EPROM und 16 KByte RAM on board, so daß sie sich mit dem mitgelieferten Maschinen-Sprache-Monitor im EPROM bereits mit einer I/O-Karte und einem Terminal betreiben läßt. Als zusätzliches Bonbon kann sie mit einem NS 32081 Floating-Point-Prozessor bestückt werden, der ihr so richtig Arithmetik-Power verleiht.

Tests mit einem variablen Taktgenerator zeigten, daß die 68er-CPU-Karte über satte Timing-Reserven verfügt. Mit einer 8-MHz-CPU war Betrieb bis zu einer Taktfrequenz von 11,5 MHz möglich. Mit einer 10-MHz-CPU ließ sich der CPU-Clock sogar bis 12,5 MHz steigern. Es sei allerdings davor gewarnt, die Karte in der Praxis mit diesen Clock-Frequenzen zu betreiben. Diese kurzen Labor-Tests liefern allenfalls eine Aussage über die Zuverlässigkeit des Systems bei 8 MHz beziehungsweise 10 MHz.

Ebenfalls eine Zuverlässigkeits-Aussage liefert das Variieren der Betriebsspannung. Das System aus CPU-, RAM-, I/O-Karte und c't-Terminal arbeitete unverändert gut mit Betriebsspannungen von 4,6 Volt bis 5,5 Volt.

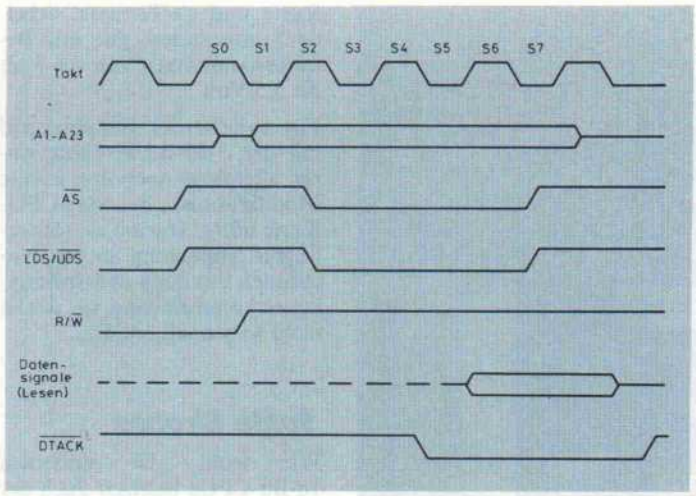
Für ein sicheres Zusammenspiel mit der c't86-RAM-Karte waren allerdings noch drei leichte Modifikationen der 68er-CPU-Karte nötig. Warum es trotz erfolgter Anpassung an den synchronen Bus noch zu Schwierigkeiten kommen kann, tja, das ist nicht so schnell erklärt.

Späte Strobes

Wie wohl alle modernen 16-Bit-CPUs besitzen auch der 8086 und der 68000 die Möglichkeit, halbe 16-Bit-Wörter, also Bytes, zu lesen und zu schreiben. Beim byte-weisen Schreiben reicht das WR-Signal der CPU natürlich nicht aus, um die richtige Worthälfte auszuwählen. Beim 8086 stehen dafür die Adreßleitung A0 und das BHE-Signal zur Verfügung. Der 68000 besitzt keine A0-Leitung, dafür aber zwei Data-Strobes (UDS und LDS), die die CPU nach Bedarf aktiviert. Da es sich eben um Data-Strobes handelt, geschieht dies erst, nachdem die Schreibdaten auf dem Datenbus gültig sind, also etwa in der Mitte des Buszyklus. Die Byte-Auswahlsignale der 8086-CPU sind dagegen von Natur aus Adressen und bereits zu Beginn des Buszyklus vorhanden.

Nun zu einem anderen Aspekt, der in diesem Zusammenhang eine Rolle spielt. Dynamische RAMs bekommen ihre Adressen in zwei Portionen serviert. Die erste Portion übernehmen sie mit dem RAS-Signal, die zweite mit dem CAS-Signal, zwischendurch muß also die zweite Adreßhälfte an die RAMs gelegt worden sein. Für einen normalen Schreibvorgang (early write) muß diese Absicht einem dynamischen RAM am Write-Enable-Eingang unbedingt vor dem CAS mitgeteilt werden. In 8086-Systemen eignen sich dazu gut die oben erwähnten Adreßsignale, die als DS0- und DS1-Signal auf dem ECB-Bus liegen und mit denen die Speicherkarten die individuellen WE-Signale für die beiden Worthälften erzeugen.

Die 68er-CPU-Karte legt auf diese beiden Schienen des ECB-Busses ihre Data-Strobes, was soll sie auch sonst tun? Die Data-Strobes sind die einzige



Lesezyklus der 68000-CPU – DTACK muß spätestens im S4-State low sein, damit die CPU keine Wartetakte einlegt und mit ihrer maximalen Geschwindigkeit arbeitet.

Möglichkeit, Auswahlsignale für byte-weises Schreiben zu erzeugen! Genau an diesem Punkt muß es zu einem Konflikt zwischen der neuen CPU-Karte und der alten 8086er-Peripherie kommen. Die RAM-Karte leitet die RAS/CAS-Signale vom MEMREQ-Signal ab, das von der 68er-CPU-Karte wiederum vom AS-Signal abgeleitet wird. Dieses Signal zeigt gültige Adressen an und liegt damit zu Beginn des Zyklus. Damit liegt aber auch noch das CAS-Signal in der ersten Zyklus-Hälfte, lange bevor die 68er-Data-Strobes die WE-Signale freigeben können, um einen eventuellen Schreibzugriff zu signalisieren. Da nun WE immer erst *nach* CAS kommen kann, leiten die RAM-Chips in bester Absicht schon mal einen Lesezyklus ein. Sie wollen also ihren Speicherinhalt an die Datenbustreiber legen, die aber bei einem Schreibzugriff nach innen gerichtet sind und ebenfalls Daten an die RAM-Chips legen wollen . . .

Kleine Änderungen . . .

Dieses Mißverständnis der RAM-Chips läßt sich durch eine kleine Änderung auf der 68er-CPU-Karte wirkungsvoll beheben. MEMREQ muß statt von AS von den und-verknüpften Data-Strobes abgeleitet werden. Dazu trenne man die auf der Bestückungsseite zu Pin 2 von IC26 führende Leiterbahn auf und lege diesen Anschluß an Pin 3 von IC29 (siehe c't 11/85, S.68). So kann MEMREQ, und damit auch RAS und CAS, erst dann aktiv werden, wenn feststeht, ob geschrieben oder gelesen werden soll und in welche Worthälfte bei byte-weisem Zugriff geschrieben werden soll.

Wer einen Blick auf das Write-Cycle-Timing des 68000 wirft, wird sich fragen, ob denn jetzt die RAM-Chips noch mitkommen, wo sich doch Adressieren und Auslesen innerhalb der nur rund 150 ns breiten Data-Strobes abspielen (8-MHz-CPU-Clock). Dies wäre in der Tat ein Grund zur Besorgnis, aber das DTACK-Signal wird ebenfalls von den Data-Strobes abgeleitet. Späte Data-Strobes bedeuten somit ein spätes DTACK. Folglich legt die CPU bei jedem Schreibvorgang einen Wartetakt ein, und die Data-Strobes sind immer rund 300 ns lang, auch beim Schreiben. Die verträdelten 125 ns beim Schreiben bedeuten kaum spürbare Einbußen an Geschwindigkeit, da nur jeder fünfte bis zehnte Zugriff ein schreibender ist, von Blocktransfers und ähnlichem einmal abgesehen.

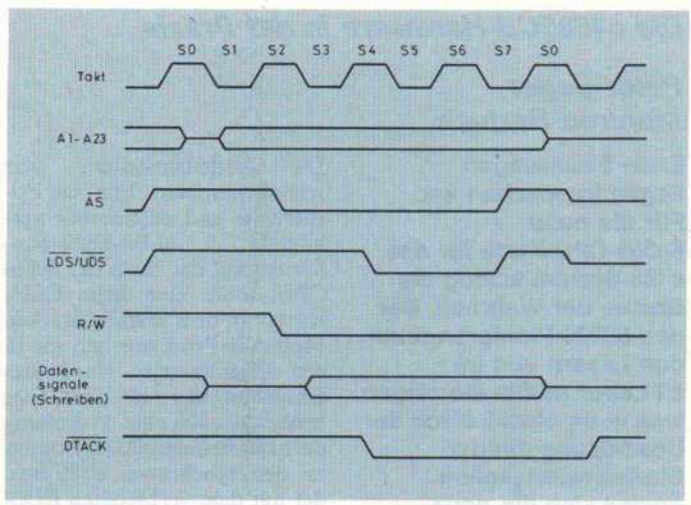
Ein weiterer kritischer Punkt bei der Kombination von 68er-Karte und 86-RAM-Karten ist die RAS-Precharge-Time vor dem Refresh. 'Nicht schon wieder', werden Z80-Kenner stöhnen. Selbst schnellste RAMs (120 ns) verlangen nach minimal 90 ns Verschnaufpause zwischen zwei RAS-Impulsen. Sie wird auf der RAM-Karte von der Phasenbeziehung zwischen dem MEMREQ-Signal und dem CPU-Takt bestimmt, was bei der 68er-Karte etwas ungünstig liegt. Aber es lassen sich leicht 62 ns Precharge-Time gewinnen, wenn man den Takt-Inverter auf der 68er-Karte herausnimmt. Am einfachsten geschieht dies durch Hochbiegen der Anschlußbeinchen 8 und 9 von IC17. Damit der Quarzoszillator nicht unmittelbar den Bus treiben muß, sollte man besser das von IC28 gepufferte Taktsignal (Pin 11) auf die ehemalige Lötstelle der Pin 8 von IC17 legen. Nach dieser Maßnahme ist man sicher im 'grünen Bereich'.

Letzte Startschwierigkeiten haben ihre Ursache in einem Layout-Fehler auf der 68er-Karte. Der Reset-Anschluß der CPU (IC1) liegt fälschlicherweise direkt am Reset-Schalter. Für einen ordnungsgemäßen Reset muß aber die Reset-Leitung zu einem definierten Zeitpunkt von low nach high wechseln. Dies ist nur gewährleistet, wenn die Leiterbahn am Reset-Pin der CPU aufgetrennt und mit dem Bus-Anschluß C 31 verbunden wird. Auch die Interrupt-Flipflops sollten ihren Reset-Impuls vom ECB-Bus beziehen. Dazu lege man die Pins 2, 5, 10 und 13 von IC19 ebenfalls an den Anschluß C31 der VG-Leiste. Nicht unbedingt störend, aber unschön ist die im Layout vergessene Verbindung von Pin 1 von IC5 nach + 5V.

. . . große Wirkungen

Nach diesen Änderungen lief eine Kombination aus 68er-Karte mit 10-MHz-CPU und einer RAM-Karte mit 120-ns-RAMs anstandslos bis zu einer Taktfrequenz von 9,5 MHz. Höhere Taktfrequenzen sind mit diesem Refresh-Timing nicht möglich.

Aber es gibt auf der RAM-Karte noch die Möglichkeit, die Flipflop-Kette, die den Refresh-Impuls erzeugt, mit dem halben CPU-Clock zu takten. Brücke 5 ist dann auf 2-3 zu jumpern (c't 8/85, S. 117). Allerdings sollte man vorher einen kleinen Hardware-Bug auf der RAM-Karte beseitigen. Das letzte Flipflop muß vom ersten



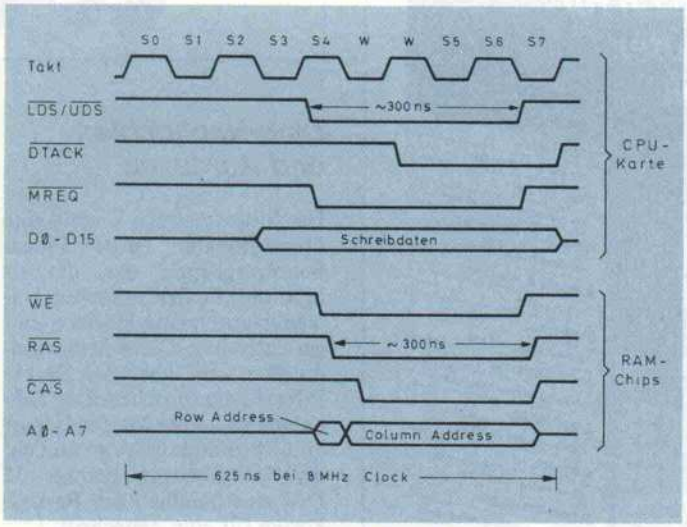
Schreibzyklus der 68000-CPU – Die Data-Strobes werden erst in der zweiten Hälfte des Zyklus aktiv und sind bei einem ungebremsten Zugriff um etwa eine Taktperiode kürzer als beim Lesezyklus.

'presettet' werden statt vom zweiten. Sonst wird ein doppelter Wait-Impuls erzeugt, der besonders beim halbierten Refresh-Takt zu Störungen führt. Also lege man Pin 10 von IC8 an Pin 6 von IC6. Nach dieser Änderung und halbiertem Refresh-Takt arbeiten CPU- und RAM-Karte bis zu einer Taktfrequenz von 11,5 MHz. Das doppelt so langsame Refresh-Timing bedingt natürlich eine größere Anzahl von Wait-States, wenn die CPU während des

Refresh auf die RAM-Karte zugreifen will.

Es gibt noch eine weitere Möglichkeit, das Refresh-Timing zu optimieren. Da es vor allem vorm Refresh-RAS-Impuls eng ist, kann es nicht schaden, den Refresh-Impuls nach hinten zu verschieben. Am einfachsten läßt sich dies durch eine Verzögerungskette aus sechs Invertern zwischen Pin 6 von IC7 und Pin 4 von IC2 erreichen. Denn das zweite Flipflop erzeugt den Refresh-Impuls. Jetzt war im c't-Labor auch noch Betrieb mit 12 MHz Taktfrequenz möglich, und zwar unabhängig von der Jumperung des Refresh-Taktes. Dies deutet darauf hin, daß für eine weitere Steigerung der Taktfrequenz die 120 ns Zugriffszeit der RAM-Chips zuzüglich der Buffer-Laufzeiten nicht mehr ausreichen, die gestreßte 10-MHz-CPU rechtzeitig zu beliefern.

Es empfiehlt sich übrigens generell, die RAM-Karte mit 120-ns-Chips zu bestücken, ganz unabhängig vom CPU-Clock. Das RAS/CAS-Timing und das Anlegen und Umschalten der Adressen geschieht nämlich sehr schnell, innerhalb von etwa 50 ns. Die Zeit, während der die RAS-Adressen anliegen, entspricht der Laufzeit von drei Buffern (IC3 und IC12), weshalb für IC14 auch ein S-Typ verwendet werden muß. Auf diese Weise erreicht man mit schnellen RAM-Chips wirklich Lese- und Schreib-Zyklen von 120 ns. Wer langsamere Chips verwenden möchte, sollte weitere Verzögerungsstrecken einbauen, etwa in Form von Gattern oder RC-Gliedern (100R bis 220R und 10p bis 33p). Verzögerungen zwischen Pin 12 von IC3 und Pin 5 von IC2 verlängern die Zeit, während der die RAS-Adressen anstehen, und verzögern den RAS-Impuls. Pin 15 von IC12 muß weiter direkt mit IC2 verbunden bleiben. Zeitglieder zwischen Pin 10 und Pin 3 von IC3 verschieben den CAS-Impuls nach hinten, und langsame RAMs haben mehr Zeit, um zu merken, daß die Adressen gewechselt haben.



Der c't68ECB arbeitet mit verlängerten Schreibzugriffen, da MEMREQ und das RAM-Timing von den Data-Strobes abgeleitet werden. Dadurch stehen den Speichern sowohl beim Lesen als auch beim Schreiben etwa 300 ns zur Verfügung.

Nichts ist vollkommen

Leider haben sich in der Beschreibung der 68er-CPU-Karte in Heft 11/85 einige Fehler eingeschlichen:

- BR7 muß genau andersherum gejumpert werden als angegeben. Nur bei EPROMs vom Typ 27512 muß BR7 auf 1-2 gesteckt werden.

- Das Oder-Gatter, das IC26 über den Enable-Eingang freigibt, ist im Layout nicht mehr vorhanden. Auf der Karte gelangt das Abschalt-Signal nach der Invertierung direkt auf Pin 19 von IC26.

- Pin 2 von IC27 liegt an Pin 21 des PALs (IC25) und Pin 1 an Pin 1 des PALs (am OFF-Signal) und nicht wie eingezeichnet an Pin 17 des PALs.

- Das Endatbuf-Signal wird von einem Und-Gatter erzeugt, und zwar von dem an Pin 4,5 und 6 von IC 29. Das dort eingetragene Oder-Gatter ist bereits an der Erzeugung des SPC-Pulses beteiligt.

Folgende Signale der c't68-CPU-Karte weichen von den strengen Vorschriften des ECB-Busses ab, führen aber nicht zu Schwierigkeiten in c't86-Systemen:

- MEMREQ läuft auch bei I/O-Zyklen durch. Dies garantiert unter allen Umständen den kontinuierlichen Refresh auf der RAM-Karte . Ein vorschriftsmäßiges ECB-MEMREQ läßt sich durch Oder-Verknüpfung der Signale ROMEN, RAMEN und CMOSEN mittels dreier Schottky-Dioden und eines Pullup-Widerstandes erzeugen.

- RD liegt statisch auf dem Bus. Zwischen aufeinanderfolgenden Lese-Zyklen wird dieses Signal nicht inaktiviert. Falls dies bei fremden ECB-Karten zu Schwierigkeiten führt, kann zum Beispiel das RD-Signal mit Hilfe des freien Oder-Gatters von IC30 (Pin 8, 9 und 10) mit AS verknüpft werden.

Schließlich schaffte es die c't-Redaktion auf insgesamt vier Fotos, mit dem Titel von Heft 11/85 beginnend, die CPU-Karte mit falsch beschrifteten EPROMs abzubilden. Auch eine Leistung. IC8 ist das EPROM für das Lower-Byte und IC9 liefert das Higher-Byte – in diesem Punkt stimmen aber Schalt- und Bestückungsplan.

Damit sollte der c't68ECB seine Kinderkrankheiten überwunden haben, und es wird Zeit, daß er erwachsen wird. Als erstes Betriebssystem wird natürlich CP/M68K verfügbar sein. Aber dabei wird es nicht bleiben.

ct

Zwei Themen – eine Ausstellung

Hobby-tronic

9. Ausstellung für Funk- und Hobby-Elektronik

COMPUTER-SCHAU

2. Ausstellung für Computer, Software und Zubehör

Dortmund

23.–27. April 1986

Die umfassende Marktübersicht für Hobby-Elektroniker und Computer-Anwender, klar gegliedert:

In Halle 5 das Angebot für CB- und Amateurfunker, Videospieler, DX-er, Radio-, Tonband-, Video- und TV-Amateure, für Elektro-Akustik-Bastler und Elektroniker. Mit dem Actions-Center und Laborversuchen, Experimenten, Demonstrationen und vielen Tips.

In Halle 4 das Super-Angebot für Computer-Anwender in Hobby, Beruf und Ausbildung. Dazu die „Computer-Straße", als Aktionsbereich, der Wettbewerb „Jugend programmiert" und der Stand des WDR-Computer-Clubs.

Ausstellungsgelände Westfalenhallen Dortmund täglich 9.00-18.00 Uhr

Ein Muß für jeden ernsthaften Programmierer.

c't-Special 1

Software-Know-how

Die wichtigsten Beiträge aus den vergriffenen c't-Ausgaben — neu bearbeitet und zusammengefaßt.

Jetzt am Kiosk*.
Preis 25,— DM.

* oder direkt ab Verlag (V-Scheck 25,— DM)

Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61.

c't-club

Club-Nachrichten und Adressen

Der Mikromodul e.V. stellt eine überregionale 'Turbo Pascal Benutzergruppe' dar, die als Medium für den schnellen Datenaustausch eine Mailbox eingerichtet hat. Diese Mikromodul-Box wird über das Datex-P-Netz angesprochen. Die Teilnahme ist an die Mitgliedschaft im Mikromodul e.V. gebunden; der Jahresbeitrag beträgt 35 DM einschließlich der Bezugskosten für eine Zeitschrift.

Mikromodul e.V.
High-Tech-Center Hamburg
Holzmühlenstr. 84
2000 Hamburg 70

Der Selbstbau-Computer-Club, eine Vereinigung der NDR-68K-Benutzer, gibt alle zwei Monate eine Club-Broschüre mit Programmen, Tips und Tricks heraus. Informationen gegen Rückporto erhält man bei folgender Adresse:

Selbstbau-Computer-Club
Dorfstraße 4
3372 Wallmoden 1

Gegenseitige Hilfestellung bei Problemen mit der Hard- und Software von Schneider-Computern will die Schneider User Group Lahn (S.U.G.L) bieten. Der Club unterstützt die Rechner CPC 464/664/6128, Joyce PCW 8256 sowie von anderen Herstellern angebotene Peripherie.

S.U.G.L
Markus Nitschke
Philosophenweg 36/3
6330 Wetzlar
0 64 41/4 86 52

Die Atari ST Gruppe Süddeutschland sucht noch Mitglieder, die sich mit der Soft- und Hardware der Atari ST-Serie beschäftigen wollen. Der Verein bietet unter anderem ein Clubinfo, Treffen, Arbeit mit der Club-Software und Seminare.

Atari ST Club
Jens Rudolph
Wiener Str. 5
8023 Pullach

Schweiz

Der 'Memotech User Club Schweiz' hat eine freie Software-Bibliothek für die Memotech-Computer aufgebaut, die jedes Mitglied des Clubs nutzen kann. Durch die Kooperation des Clubs mit einem englischen Verein ist das Angebot der Bibliothek sehr umfangreich.

Memotech User Club Schweiz
Christian Wöhlbier
Hardstrasse 13
CH-4711 Aedermannsdorf

Der Epson-User-Club gibt alle zwei Monate eine 50 Seiten starke Zeitschrift heraus und veranstaltet jeden Monat ein Treffen. Der Club mit dem Namen EPSILOG ist in mehreren Mailboxen vertreten und bietet eigene Software an.

EPSILOG
Postfach 185
CH-8704 Herrliberg

Kontakte

Zwecks Erfahrungsaustausch mit MC-CP/M-Computer-Anwendern im Postleitzahlgebiet 4000 sucht Kontakt:

P.-A. Marx
Rheinberger Str. 319
4132 Kamp-Lintfort

Kontakt zu Nachbauern des c't86ECB sucht:

Jens Harbott
Flemingweg 17
4300 Essen 1

Suche Kontakt zu c't68000-Selbstbauern im Raum Reutlingen-Stuttgart:

Gerhard Durant
Riedlinger Straße 17/32
7410 Reutlingen

Suche Kontakt zu Spectra-Video-Usern beziehungsweise Clubs, die unter CP/M 2.2 ein GO-Programm kennen und wissen, wo es das gibt.

Martin Leidig
Obere Stadt 3
8650 Kulmbach

COMMODORE Sonderangebot

... solange Vorrat reicht!

Commodore 610 Personalcomputer

Prozessor: 6509 (8 Bit).

Speicher: 128 KByte RAM; 24 KByte ROM.

Schnittstellen: Eingebaute RS-232-C-Schnittstelle; Tonausgang; IEEE-488 BUS zur Ansteuerung von Massenspeichern, Druckern und Zusatzgeräten. Die Vernetzung von mehreren Geräten ist möglich.

Sonstiges: 80-Z.-Darstellung; Tastatur mit 94 Tasten, separatem Cursor und Zehnerblock. BASIC 4.0 Betriebssystem (erweiterbar um 24 KB). Dreistimmiger Soundsynthesizer 6581. Monitorausgang NTSC/BAS (1 V/75 Ohm).

Lieferumfang: Netz, Videokabel, Handbuch (dt.).

Neu, ungeprüft, ohne Garantie.

Best.-Nr. 9911018 ..... 298,— DM

Völkner electronic

Postfach 5320
33 Braunschweig
Telefon (05 31) 87 62-111
Telex 9 52 547

Bitte fordern Sie unsere kostenlose SONDERLISTE 1/86 an.

HARDWARE UND SOFTWARE
LÖSUNGEN FÜR
TECHNIK UND WIRTSCHAFT
AUF MIKROCOMPUTERN

olivetti Fachhändler

M24, Bildschirm, Tastatur, MS-DOS, **640 KB RAM**	
mit 1 FD 360 KB, **10.5 MB Harddisk**	9.764,–
mit 1 FD 360 KB, **21 MB Harddisk**	10.323,–
mit 1 FD 360, **21 MB Hardd., 20 MB Streamer**	13.800,–
Für 720 KB Laufwerke gelten die gleichen Preise!	
128 KB RAM Speichererweiterung	70,–
512 KB RAM Speichererweiterung	253,–
BUS-Converter für M24	399,–
8087-2 (8 Mhz) Co-Prozessor	645,–
2. V.24 Schnittstellenkarte	253,–
HDU/STREAMER Erweiterungen für alle PC's	
● 10.5 MB HDU incl. Controller	1.875,–
● 21 MB HDU incl. Controller	2.435,–
● 42 MB HDU incl. Controller (35 ms)	6.785,–
● Ext. Subsystem 21 MB HDU, 20 MB Streamer	6.600,–
● Ext. 20 MB Cassetten-Streamer	3.919,–
● 10 MB LOW-Cost Streamer	2.668,–

star Vertragshändler:

Druckerkabel im Preis inbegriffen!	
SG-10, 120 Z/s, Schönschrift	1.090,–
SD-10, 160 Z/s, IBM-PC/ASCII umschaltbar	1.456,–
SR-10, 200 Z/s, Autom. Einzelblatteinzug	1.965,–
SG-10 anschlußfertig für Commodore C64	1.191,–
Alle Drucker in 15" mit 16 KB Zeichenpuffer	
Vollautomat. Einzelblatteinzug f. alle Drucker	992,–

Maßgeschneiderte Softwarelösungen
Nachrichten- und Datenkommunikation
Eigene Servicewerkstatt
für STAR-Drucker und OLIVETTI-M24

Akustikkoppler-Modem CTK 2000	1.115,–
V21/V23/300-/600-/1200-Baud/BTX	
LOW-Cost Akustikkoppler CDI 300	295,–
TELEX-Interface MS-DOS, CP/M, FTZ-Nr.	4.450,–
V24 Schnittstellentester	490,–
4-Farb DIN A3 Plotter (HP 7475 kompatibel)	1.990,–
Umschalter (V24/Centronics) 1xEin/2xAus ab	253,–
SOFTWARE – DEUTSCHE VERSIONEN	
WORDSTAR 2000 deutsch	1.680,–
OPEN ACCESS für M24 deutsch	1.380,–
DATEX-P Kommunikationspaket/Mailbox	336,–

Alle Preisangaben **inklusive** Mehrwertsteuer

SYSTEMHAUS MÜNCHEN
SPP PIPER & PARTNER GMBH
8 München 19 · Romanstr. 64 · Tel. 089/17 44 77 o. 17 06 09
Tx. 522 947 hpmuc d – TTX 898 861 = PiprMuc

Der **Eprommer** für **Apple //e, Apple II,** kompatible und Schneider **CPC 464/664**

Uhiverseller EPROM-Programmer 4003

■ Programmiert alle gängigen EPROM-Typen (z.B.: 2716,-32,-64,-128,2508,-16,-32,-64) ■ Voll menügest. Software auf Diskette/Kassette ■ Kein Schalten, Stecken oder Löten nötig ■ Programmiersp. wird im Gerät erzeugt ■ Verbindung zum Rechner über Flachbandkabel ■ Rote und grüne Leuchtdiode zur Betriebs-Art-Anzeige ■ Kompl. mit 28 poligem Textool-Sockel ■ CPC-Version mit Interface-Karte und durchgeführtem Expansionsport ■ Gleichzeitiger Anschluß der Floppy möglich ■

Preise für Apple : Fertiggerät DM 269,50 ■ Bausatz DM 219,-
für Schneider CPC: Fertiggerät DM 289,50 ■ Bausatz DM 239,-

CPC-EPROM-Karte 64 KByte

Die ideale Ergänzung für jeden Schneider CPC 464/664

■ Wahlweise bestückbar mit 2 - 64 KByte EPROM-Kapazität ■ Arbeitet mit den EPROM-Typen 2716,-32,-64,-128 ■ Durchgeführter Erweiterungsbus (Floppy kompatibel) ■ Autostart von BASIC- und/oder Assembler-Programmen ■ Komplett mit umfangreicher und komfortabler Software ■ Gleichermaßen für Profis und Einsteiger geeignet ■

■ Fertiggerät DM 249,50 ■ Bausatz mit Anleitung DM 219,50 ■

80 Zeichenkarte + 64 KByte RAM für Apple //e

■ 80 gestochen scharfe Zeichen/Zeile ■ Plus 64 KByte RAM ■ Ermöglicht Double Hires Grafik (560 × 192 Punkte, 16 farbig) ■ 100% Apple //e kompatibel ■ Läuft problemlos unter CP/M, Pascal, DOS, ProDOS ... ■ Vergoldete Steckerleiste ■

Geprüfte Platine plus Demo Disk und Beschreibung DM 164,50 ■ Bausatz DM 135,- ■ Leerplatine mit Anleitung DM 59,- ■

Speedy 100-80

■ 100 Zeichen / Sek. schnell ■ FX80 kompatibel ■ Bis zu 142 Zeichen pro Zeile ■ Optionaler Druckerpuffer ■ grafikfähig ■ Bidirektional Druckweg optimiert ■ Centronics-Schnittstelle ■ Internationale Zeichensätze ■ Eingebauter Selbsttest ■ NLQ ■ Friktionswalze und Traktorantrieb serienmäßig ■

■ Kompl. mit deutschem und englischem Handbuch DM 739,- ■
■ Zusätzlicher Druckerpuffer 2K DM 25,- ■ 4K DM 50,- ■

Druckerkabel CPC 464/664 DM 35,- ■ CPC 6128 DM 45,- ■

Preise inkl. Mehrwertsteuer. Alle Artikel ab Lager lieferbar.

DOBBERTIN
INDUSTRIE-ELEKTRONIK
Brahmsstraße 9, 6835 Brühl, Tel.: (06202) 71417

"Design your own chip"

PAL-PROGRAMMER für APPLE II (e)

Vollständiges Paket für den Entwurf und die Programmierung von PAL ICs:

- Apple Slotkarte mit Zero-Force Fassung
- programmiert 20- und 24-pol. PALs
- Schutz Ihrer PALs vor Kopie und Nachbau durch Brennen der Security-Fuse

Menügesteuertes Autostart-Softwarepaket:

- Ohne Programmier- oder Computerkenntnisse bedienbar
- Eingabe, Editierung und Speicherung Ihres PAL-Entwurfes
- PAL-Assembler übersetzt die gewünschten Logikfunktionen in das entsprechende Fuse-Pattern (Programmiermuster)
- Brennen, Lesen, Kopieren auf Knopfdruck
- Screen-Editor für Fuse-Pattern
- ausführliche Dokumentation
- **DM 1103,50** inkl. 14% MwSt. **1 Jahr Garantie**

NUCLEAR INTERFACE
Datentechnik für Strahlungsmeßgeräte GmbH
Goldstraße 64, 4400 Münster
Telefon: 02 51/27 35 85

Händleranfragen erwünscht!

Neu:

CP/M-68K

für den c't68000 995 DM

CP/M-68K-Programmpaket von Digital Research mit HSP-BIOS für den c't68000. Lieferumfang (unter anderem): C-Compiler, Assembler, Linker, Debugger, zeilenorientierter Editor (ED), Formatierer, Backup-Programm, CEDIT-Demoversion, CP/M-Z80-Demoversion. Mitgeliefert werden die Original-Handbücher von Digital Research (User's Guide, Programmer's Guide, System Guide und C Language Programming Guide) sowie eine Bedienungsanleitung für das HSP-BIOS und die zusätzlichen Dienstprogramme.

Das HSP-BIOS unterstützt standardmäßig 5,25-Zoll-Laufwerke mit 2 × 80 Spuren und 1024 Byte/Sektor (Kapazität 800 KByte) und zwei weitere Formate. Außerdem ist eine RAM-Floppy implementiert.

So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck über die Bestellsumme zuzüglich DM 3,— (für Porto und Verpackung) bei oder überweisen Sie den Betrag auf eines unserer Konten.

Schecks werden erst bei Lieferung eingelöst. Wir empfehlen deshalb diesen Zahlungsweg, da in Einzelfällen längere Lieferzeiten auftreten können.

Bankverbindungen:
Postscheckamt Hannover, Kt.-Nr. 93 05-308
Kreissparkasse Hannover, Kt.-Nr. 000-019968
(BLZ 250 502 99)

Ihre Bestellung richten Sie bitte an:

Verlag Heinz Heise GmbH
Postfach 61 04 07
3000 Hannover 61

Computerplatine HCP-80 — PSI-80 kompatibel —

Z80A-CPU (4 MHz), 2 x V24-Schnittstelle, 1 x Centronic-Schnittstelle, Tastaturanschluß parallel/8 Bit, Bildschirmanschluß BAS oder TTL, Anschluß für 4 Laufwerke (5/8-Zoll gemischt), ECB-Bus, 128 KB-RAM, Vollgrafik bis 768 x 680 Bildpunkte.

DM 2800,— incl. MWSt.

angepaßtes CP/M 2.2 **DM 450,— incl. MWSt.**

DIN-A3-Flachbettplotter HPX-84

Auflösung und Wiederholgenauigkeit: 0,1 mm
Schreibgeschwindigkeit: 120 mm/sec
Schnittstelle: Centronic

Software: 18 Befehle für Vector, Kreis, Rechteck, Maßstab, Text (ASCII), Geschwindigkeit, Randbegrenzung.

Siehe auch Testbericht in "Computer Persönlich 17/85"

HPX-84-Flachbettplotter
Bausatz **DM 1198,— incl. MWSt.**
Fertiggerät **DM 1398,— incl. MWSt.**

Firma Peter Habersetzer
Hard- und Software-Entwicklungsbüro
St.-Jakob-Straße 8B
8121 Polling

WIR KOSTEN ZWAR AUCH,
ABER ÄRGERN SIE NICHT!

Spezialisten sind wir dadurch geworden, daß wir unseren Kunden bei der Installation, Wartung und im Servicefall unserer Massenspeicher-Systeme jederzeit zur Verfügung stehen.
Ständig neue Gegebenheiten vor Ort vermitteln uns viel Erfahrung, so daß wir in der Beratung manchen Kunden vor Fehlentscheidungen bewahren konnten.
Fehlentscheidungen sind ärgerlich teuer und halten Sie von Ihrer eigentlichen Tätigkeit ab – Tätigkeiten, die Ihren Erfolg sichern.

Wir erweitern Ihren PC mit speziellen Massenspeichern.
– Jeder Spezialist an seinen Platz!

HETRON

HETRON GmbH, Pfingstrosenstr. 29, 8000 München 70, Tel.: 089/716007-9, Tlx: 5214372he d

ATARI®

BNT — der ATARI-Spezialist

Jedes Softwareprogramm für 'ST' vorrätig,
über 200 Artikel für 'ST' auf Lager.
SUPERPREISE, z. B. Thomson Farb-Monitor
für 'ST' 789,— DM.

Fordern Sie unsere kostenlose Preisliste an.

BNT Computer GmbH
Marktstr. 48
7000 Stuttgart 50
Tel. 07 11/55 83 83

Turbo-Tutor

Heimsoeth-Software
Fraunhoferstr. 13
8000 München 5

MS/PC-DOS, CP/M-86, CP/M-80
Preis: 104,86 DM

Wollen Sie in Pascal programmieren und haben sich aus diesem Grund für Turbo-Pascal entschieden? Wenn Sie dann aber bemerken, daß sich dieser Pascal-Dialekt doch ziemlich von dem unterscheidet, der an den meisten Instituten gelehrt wird, muß also ein Lehrbuch her. Es muß dem angehenden Turbo-Programmierer die Eigenheiten dieses Pascals vermitteln. Mit dem Turbo-Tutor befindet sich ein solches Lehrbuch nebst Begleitdiskette auf dem Markt.

Es beginnt mit der Geschichte der Programmiersprachen im allgemeinen und Pascal im besonderen. Bereits das nächste Kapitel konfrontiert den Leser jedoch mit der Entwicklung eines Programms.

Dabei wird dem Leser der Unterschied zwischen Variablen und Programmanweisungen 'beigebracht', und ehe er sich versieht, hat er ein funktionierendes Programm im Speicher seines Rechners. Nachdem geklärt wurde, welche Arbeit Turbo-Pascal verrichten soll, widmet sich ein weiteres Kapitel dem Start von Compiler und Editor.

Nach diesen einführenden Abschnitten beginnt der zweite Teil des Turbo-Tutors, der 'Leitfaden für den Pascal-Programmierer'. Beginnend mit der Vereinbarung von Variablen und Konstanten lernt der Leser in diesem Teil, was die Sprache Turbo-Pascal ausmacht. Programmkopf, Programm und vor allem Funktionen und Prozeduren werden analysiert und durch Beispiele untermauert. Besonders ausführlich werden die Dinge behandelt, die mit Feldern und Zeichenketten zu tun haben. In diesem Bereich sind die Unterschiede zum Wirthschen Pascal aber auch besonders groß.

Der Leitfaden schließt mit einer Reihe von Ausführungen über Dateien und Record-Typen ab. Die Ein- und Ausgabe von und zum Massenspeicher wird ebenso ausführlich behandelt wie die übrigen Turbo-Besonderheiten.

Der dritte und letzte Teil des Tutors trägt den Titel 'Fortgeschrittene Programmiertechniken'. Am Beispiel von den Betriebssystemen CP/M und MS-DOS werden verschiedene Möglichkeiten aufgezeigt, mit Turbo-Pascal auf Betriebssystem-Routinen zuzugreifen und per Pascal an die gewünschten Informationen zu gelangen.

Die Begleitdiskette zu dem Buch enthält alle im Tutor vorgestellten Programme. Als Zugabe befindet sich noch ein neuentwickeltes Druckprogramm zur Ausgabe von Pascal-Quellcodes auf der Diskette.

Turbo-Tutor ist gut geeignet, um Turbo-Pascal kennenzulernen. Der Preis übersteigt einschließlich der Diskette nicht den von anderen guten Lehrbüchern. Der recht lockere, persönliche Stil, in dem der Tutor verfaßt ist, erleichtert das Lesen und Nachvollziehen von Programmen.

PG

PROLOG 64

Brainware GmbH
Kirchgasse 24
6200 Wiesbaden

Diskette für C64
Preis: 289,00 DM

Das Angebot an Programmiersprachen für den C64 ist nicht sehr breit. Mit PROLOG 64 gibt es jetzt eine Implementierung der gleichnamigen Sprache. Zum Lieferumfang gehören eine Diskette mit dem Interpreter und verschiedenen Beispielprogrammen sowie ein Handbuch.

Nach dem Start ist das Programm im Interpreter-Modus, in dem man Fragen an das System stellen kann. Zur Eingabe von Funktionen steht 'consult (FN)' zur Verfügung; mit 'consult(user)' kann man Eingaben über die Tastatur vornehmen.

Der Interpreter von PROLOG 64 liegt unter den ROMs, so daß die kompletten 64 KByte Arbeitsspeicher für den Anwender verfügbar sind.

Was für Pascal der 'Report' von Jensen/Wirth ist, ist für Prolog das Buch von Clocksin/Mellish. Es enthält den 'Edinburgh Stan-

dard', an den sich auch PROLOG 64 hält. Einschränkungen betreffen nur einzelne Rahmenwerte und Details bei Listen.

Dagegen gibt es eine Reihe von C64-spezifischen Erweiterungen. So ist eine Turtle-Grafik implementiert, die durch zusätzliche Bibliotheksfunktionen noch erweitert wird. Die Funktionen können auch Dreiecke, Kreise oder Balkendiagramme zeichnen. Sprites müssen mit POKEs gesteuert werden.

Mit 'sound' kann man Töne beliebiger Höhe, Dauer und Klangfarbe erzeugen; in einer Bibliothek sind Beispiele zur Tonerzeugung vorhanden.

Außerdem sind einige Prolog-Programme vordefiniert, die zur Listenmanipulation dienen. In den Bibliotheken findet man jedoch auch einige komplexere Programme beziehungsweise Funktionen.

Mit 'see' und 'tell' kann man Ein- und Ausgaben umleiten. Somit kann das System auch Files verwalten, was zum Beispiel zur Speicherung von Daten nützlich ist.

Für die Fehlersuche sind die Funktionen 'spy' und 'nospy' gedacht. Sie beziehen sich auf ein Prädikat und stoppen das System bei der Ausführung dieses Prädikats. Die Funktionen entsprechen somit den bekannten 'Breakpoints'.

Da Prolog eine Interpretersprache ist, kann man mit dem System 'herumspielen', um mit der Sprache vertraut zu werden – typische Anfängerfehler nimmt das System nicht übel.

Bei der Arbeit zeigt sich dann auch, daß das umfangreiche Backtracking bei der Auswertung der Regeln den C64 stark beansprucht. Die Antworten dauern zwar schon etwas lange, insgesamt ist die Geschwindigkeit jedoch annehmbar.

Es wird eben nicht nur hier sichtbar, wo die Grenzen von PROLOG 64 liegen. Man kann auch von der Hardware des C64 nicht erwarten, daß ein Expertensystem mit umfangreichem Wissen auch noch schnell ist. Zur richtigen Anwendung ist wohl mindestens ein PC nötig. Zum Erlernen und Verstehen von Prolog reicht die Implementation auf dem C64 sicher aus. PROLOG 64 überzeugt durch seine einfache Bedienung, nützliche Zusatzfunktionen und annehmbare 'Performance'.

Das Handbuch besteht aus 50 Seiten Beschreibung sowie aus Listings der mitgelieferten Beispielprogramme. Dabei handelt es sich um eine reine Implementationsbeschreibung; zum Erlernen von Prolog wird auf geeignete Bücher verwiesen. Zwei Kapitel sind zwar der Programmierung in Prolog gewidmet, es werden jedoch nur dessen Möglichkeiten angerissen.

Der größte Teil beschreibt die Syntax, die eingebauten Funktionen sowie die Bedienung von PROLOG 64. Die Erläuterungen zu den Funktionen sind leider nicht alphabetisch geordnet, was beim Nachschlagen etwas hinderlich ist. Der Autor des Handbuchs geht gesondert auf spezielle C64-Erweiterungen ein. Die Beschreibungen sind zwar knapp, reichen aber aus.

Die Listings der Beispielprogramme sind identisch mit der Software auf Diskette, die Kommentare beschränken sich nur auf das Nötigste. Es sind meist nur Angaben zum Aufruf der Funktionen beziehungsweise Prädikate vorhanden; auf die Funktionsweise geht der Autor nicht ein. Die Programme haben zumeist reinen Bibliothekscharakter, sie stellen also Funktionen zur Verfügung, mit denen man dann als 'Black Box' arbeiten kann. Die weiteren Programme demonstieren die C64-Erweiterungen oder sind Implementationen von Standard-Lösungen in Prolog.

Wer eine außergewöhnliche Programmiersprache kennenlernen will, ist mit PROLOG 64 gut bedient. Der Preis ist für C64-Verhältnisse zwar hoch, an 16-Bit-Produkten gemessen jedoch sehr klein. Dafür erhält man ein ausgereiftes Produkt, mit dem auch Anfänger wenig Schwierigkeiten haben sollten.

Die Dokumentation ist ausreichend, ersetzt aber in keinem Fall weitere Literatur. Es handelt sich praktisch um ein gutes 'Reference-Manual', das kaum Fragen bezüglich der konkreten Implementation offenläßt.

Es macht ganz einfach Spaß, mit dem System zu experimentieren. Dabei ist sehr angenehm, daß es sich um ein 'sicheres' System handelt. Man braucht also nicht in der ständigen Angst vor einem Systemabsturz leben. RT

DV-Büromaschinen Peter Schmoll
Badestraße 1 — 3130 Lüchow 1 — Telefon (05841) 5296

Hardware — Software neu und gebraucht

Second Hand Hardware

PC, 2 Laufwerke, 256KB, Monochrom-Monitor, Tastatur ab (auch neu) 3900,— DM

XT, 1 Laufwerk, 256KB, 10MB-Festplatte, Monochrom-Monitor, Tastatur ab (auch neu) 5100,— DM

IBM-AT 02, 1 Laufwerk, 1,2MB, 20MB-Festplatte, 512KB, Monochrom-Monitor, Tastatur nur 11800,— DM

Second Hand Software

MS-Word 2.0 nur	630,— DM
Wordstar 3.4 nur	420,— DM
dBase III nur	635,— DM
dBase-Compiler	1300,— DM
Turbo-Pascal 3.0	110,— DM
C-Compiler ab	150,— DM
Fibu ab	640,— DM
Faktura ab	760,— DM

u.v.a.

Gebrauchte ORIGINAL-Software aus Betriebsauflösungen usw.

Fordern Sie unsere Preisliste an!
Erfragen Sie Tagespreise!

Matrix-Printer	ab 450,— DM	Monitore	ab 430,— DM
Typenrad-Drucker	ab 670,— DM	u.v.a.	
Tastaturen	ab 280,— DM		

Alle Gebrauchtgeräte mit 6 Monaten Garantie
10 Tage Rückgaberecht (außer für Software und Verbrauchsmaterial)
Lieferung gegen Vorkasse oder NN

Markendisketten 10 St. ab 34,— DM

Bieten Sie uns Ihre gebrauchte Hard- und Software an!

CP/M SOFTWARE

Softwareliste (CP/M — CDOS — Cromix) 3/84

		Manual	Komplett
Nevada COBOL	Compiler	DM 29.—	DM 129.—
Nevada FORTRAN	Compiler	DM 29.—	DM 129.—
Nevada PILOT	Interpreter	DM 29.—	DM 129.—
Nevada BASIC	Interpreter	DM 29.—	DM 129.—
Nevada EDIT	Editor	DM 29.—	DM 129.—
Nevada Pascal	Compiler	DM 29.—	DM 129.—
C/80 Compiler			DM 189.—
C/80 MathPak (Long/Float C/80)			DM 99.—
LISP/80 Interpreter			DM 189.—
RATFOR/80 FORTRAN preprocessor			DM 189.—
ZENCALC/80 Spreadsheet			DM 189.—H
MYCALC/80 Spreadsheet			DM 199.—
JRT Pascal 3.0 Compiler			DM 199.—
Epic Supervyz (Menue System)			DM 299.—
Clip (Unix-Shell für CP/M)			DM 199.—Z
Clip-Tools (Unix-Utilities)			DM 99.—Z
Clip-Crypt			DM 99.—Z
M/PC CP/M 2.2 Source Generator			DM 199.—
CP/M User Group Diskette/Vol.			DM 22.80
SIGM User Group Diskette/Vol.			DM 22.80
TDRIVE Ram-Disk driver f. CDOS		DM 19.—	DM 249.—C
TSPOOL Ram-Printer-Spooler CDOS		DM 19.—	DM 99.—C
TCPM-C (Netzplan/SBASIC)		DM 19.—	DM 99.—C
TCPM (Netzplan/MBASIC)		DM 19.—	DM 99.—
TURBO-Pascal			DM 219.—

Lieferung ab Lager bzw. je nach Diskettenformat innerhalb zwei bis vier Wochen. Alle Preise inkl. Mehrwertsteuer, Porto u. Verpackung! Alle vorherigen Listen verlieren ihre Gültigkeit. (Z=benötigt Z80 CPU, C=benötigt CDOS Betriebssystem, CX=benötigt Cromix Betriebssystem, H=benötigt Heath H 19 oder kompatibles Terminal).

TESCO GmbH
P.O.Box 10

West Germany
8714 Wiesentheid
Tel.: 09383/1237

NEU P.C. QUELLE NEU

IBM-kompatibel

— PC-XT-256 K, color/gr.
1 Laufwerk 360 KB.
und Netzteil 130 W.

1 995,—

— PC-XT-256 K, color/gr., 2 Laufwerken 360 KB,
Netzteil 130 W, Software

2 195,—

— PC-12 w. oben, RS232 par. clock game I/O 2395,—

OKIDATA Microline ML 192 ..... **DM 1 585,—**
OKI 20 ..... **DM 898,—**
Roland-Plotter DXY 880 ..... **DM 3 275,—**

PREISE INKL. MWST.

NEU KÖLNERSTRASSE 49
4000 DÜSSELDORF 1
TEL: 0211 – 161664 NEU

Wie erfassen Sie Ihre Daten?

(Wenn Sie noch immer Papier und Bleistift benutzen, sollten Sie noch heute mit uns sprechen!)
Wir bieten Ihnen mobile und stationäre Datenerfassungssysteme für den Einsatz von Strichcodes. Zum Beispiel:

- **Handterminal T 100/T 200**
  Mikrocomputer (batteriebetrieben) für den mobilen Einsatz. 32stelliges LC-Display. Programmspeicher mit Firmware für applikationsabhängiges Anwenderprogramm. Datenspeicher zum Zwischenspeichern der erfaßten Daten. Datenübertragung akustisch über jedes Telefon oder über RS 232C-Schnittstelle. Dateneingabe mit Strichcode-Lesestift oder Tastatur.
- **Strichcodedecoder BD 100**
  Stationäres Strichcode-Lesesystem mit Decodersoftware für alle gängigen Strichcodes. Datenübertragung über RS 232C-Schnittstelle oder Modem. Netzwerkverbund über Konzentrator. Dateneingabe über Lesestift oder starre Abtastoptiken. Spannungsversorgung über einfaches Steckernetzteil.

Einsatzmöglichkeiten: Betriebs- und Fertigungsdaten, Bestelldaten, Inventur, Warenein- und Ausgang.
SOFTWARE SYSTEMS — Wir bieten Ihnen Systemlösungen!

SOFTWARE SYSTEMS Vertriebs-GmbH
Hildesheimer Str. 15, 3000 Hannover 1
Telefon (05 11) 85 51 34
Treibersoftware für div. PC's vorhanden!

MAXAM-Assembler

Ralf Probst EDV-Systeme
Friedrich-Ebert-Straße 14
4100 Duisburg 17

Schneider CPC 464/664
Kassette: 69,00 DM
Diskette: 109,00 DM
ROM-Modul: 239,00 DM

Assembler scheinen, genauso wie Textverarbeitungsprogramme, zu den Lieblingsobjekten von CPC-Programmierern zu gehören. Wie wäre es sonst zu erklären, daß es inzwischen eine ganze Reihe verschiedener Assembler und Maschinensprache-Monitore gibt?

Der MAXAM-Assembler der britischen Firma Arnor sticht allerdings hervor: zuerst gibt es ihn in einer Version als ROM-Modul, das nur auf den Expansion-Port des Computers aufgesteckt werden muß und dann ohne Ladevorgang das Programm zur Verfügung stellt.

Zum Test hatten wir die Diskettenversion und das ROM-Modul. Beide sind identisch; bis auf den belegten Speicherplatz und die Ladezeit. Wird während der Arbeit jedoch der Speicherplatz knapp, kann man entweder nur den Editor oder nur den Assembler und Monitor von Diskette laden.

Ruft man den Assembler mit dem RSX-Befehl MAXAM auf, zeigt er sofort das Hauptmenü. Neben der Möglichkeit RSX-Befehle einzugeben, bietet er verschiedene Monitorfunktionen: Disassembler, Speicher anzeigen oder editieren, Erweiterungs-ROMs in den Adreßbereich kopieren, Blöcke vergleichen, Maschinencode verschieben, Speicher mit Bytes füllen. Die AMSDOS RSX-Kommandos kann man nun auch erheblich bequemer eingeben: sie sind im Aufbau an CP/M-Anweisungen angelehnt.

Sobald man die Taste 'T' für 'Text-Editor' drückt, erscheint ein neues Menü, das eine Vielzahl von Editor-Funktionen anbietet. Mit 'E' für 'Edit Text' gelangt man in einen Full-Screen-Editor, der nicht nur als Quellcode-Editor geeignet ist – man kann damit auch Briefe schreiben. Der Text wird wahlweise mit 40 oder 80 Zeichen pro Zeile dargestellt. Damit ist aber noch die maximale Zeilenlänge begrenzt; eine Textzeile kann beliebig lang sein. Wenn es sein muß, bis zu etwa 42000 Zeichen, also der maximalen RAM-Kapazität. Nebenbei bemerkt: Die Zeichenausgabe und das Bild-Scrolling arbeiten um ein Vielfaches schneller als die ROM-Routinen.

Der Assembler wird aus dem Menü des Editors heraus mit 'A' aufgerufen. Das Programm ist sehr schnell und komfortabel. Es ist offensichtlich so ausgelegt, daß es den Quellcode von möglichst vielen verschiedenen Assemblern 'verstehen' und verarbeiten kann. Zum Beispiel versteht der Assembler DEFB, DB, DEFM, BYTE und TEXT. Labels können beliebig lang sein, wobei alle Stellen signifikant sind. Außerdem gibt es eine Vielzahl von Pseudobefehlen zur bedingten Assemblierung, zum Einbinden von Quellcode-Dateien auf Diskette oder Kassette, zur Erstellung von CP/M-COM-Files, zur Datenübergabe zwischen BASIC und Maschinenprogrammen und zur Formatierung des Drucker-Listings beim Assemblieren.

Eine nützliche Eigenschaft von MAXAM ist, daß der von Amstrad-Programmen nicht verwendete Restart 6 als Einsprung für die Breakpoint-Routine dient. Sobald dieser Restart-Befehl oder das Pseudo-Kommando BRK auftaucht, verzweigt das Betriebssystem in eine Routine, die die Registerwerte ausgibt und auf einen Tastendruck wartet.

Daneben bietet MAXAM noch eine BASIC-Erweiterung. So kann man mit den Befehlen ASSEM und ASSEMBLE BASIC- und Assembler-Programme gemischt verwenden. So können zum Beispiel die ersten Zeilen aus BASIC-Anweisungen bestehen, gefolgt von dem Befehl ASSEMBLE und dem Quellcode des Maschinenprogramms in REM-Zeilen. Beim Programmlauf wird dann der Maschinencode übersetzt und im Speicher abgelegt. Der RSX-Befehl ASSEM ermöglicht es, die Ausgabe eines Assembler-Listings zu unterdrücken, wodurch die Übersetzung sehr stark beschleunigt wird.

Alles in allem ist MAXAM bislang wohl das perfekteste und vollständigste Entwicklungspaket für Maschinenprogramme, das momentan am Markt ist. Die Bedienung dieser Software ist so einfach, daß man schon nach kurzer Zeit das Handbuch nicht mehr benötigt. MK

Z80 - 8085 - NSC800
Emulatoren

Die preisgünstige Lösung für anspruchsvolle Emulation

DM 2736,—

Kurzfristige Lieferung
ab Lager Rosenheim durch

Schwarz & Müller KG
Buchenweg 5
8209 Stephanskirchen
Tel. 0 80 31/7 11 62

Komplett mit Benutzerführung, 32 Kbyte Editier-RAM, Emulator-Funktion, V.24, Netz/Akku-Betrieb und den Funktionen, die in der 3000-DM-Klasse einfach üblich sind, für **DM 1.980,-- o.MwSt.**

Meßtechnik Dr.-Ing. R. Ranfft
Dörpfeldstr. 15. 5657 Haan 2. Tel.: (02104) 62827

Pascal

McGraw-Hill

Programmiertechnik für Fortgeschrittene
Paul A. Sand

NEU

Das Buch führt den Leser über den Stand von Einführungen in diese weit verbreitete Programmiersprache hinaus, zu einem tieferen Verständnis der Zusammenhänge und zur selbstverständlichen Beherrschung von Pascal-Programmiertechniken. Es wendet sich an den bereits erfahrenen Programmierer oder Systemtechniker, der viele der Beispiele mit oder ohne Veränderung in eigene Programme einbauen kann. Die Listings dokumentieren einige der besten Seiten von Pascal in der schulischen und beruflichen Anwendung. Tips zur interaktiven Eingabe, zum modularen Programmaufbau, zur Bildschirmmanipulation und zum Aufbau eines eigenen elektronischen pascalc-Arbeitsblattes (einem Kalkulationsprogramm) und anderer, auch kommerziell verwendbarer Programme machen PASCAL - PROGRAMMIERTECHNIK FÜR FORTGESCHRITTENE zum wertvollen, ja unverzichtbaren Begleiter auf dem Weg zu besseren Pascal-Programmen.

ca. 380 Seiten, ca. DM 35,-

Fordern Sie unseren Gesamtprospekt an!
Coupon ausschneiden und einsenden an:
McGraw-Hill Book Company GmbH
Lademannbogen 136, 2000 Hamburg 63
In der Schweiz:
Industriestraße 2, CH-6282 Hitzkirch

Bitte senden Sie mir den Gesamtprospekt COMPUTER-LITERATUR

Name _____

Anschrift _____

miniCON-52

Mit 8052 AH-BASIC das kleinste komplette Mikrocontroller-Entwicklungssystem.

- CPU-Platine für Intel MCS 51-Familie
- PIO 8255 mit max. 24 I/O-Leitungen
- EPROM-Programmer
- 8 K RAM, 2 x 28 pol. Jedecsockel
- max. 32 K EPROM, Textoolsockel opt.
- RS 232 Schnittstelle + Printerschnittstelle
- 70 x 90 mm² WRAP-Bereich
- Leicht an ECB-Bus anzuschließen
- Europakarte 100 x 160 mm
- statt EPROM opt. NVRAM
- Umfangreiches Begleitmaterial incl. 8052 AH-BASIC-Manual

Kompl.-Gerät DM 398,- + MWST

PHYTEC

PHYTEC Meßtechnik, Dipl.-Phys. Karl Neubecker
Donnersbergstr. 32, 6500 Mainz 42, Tel. 0 61 31/50 88 71

ASC-COMPUTER-ELEKTRONIK

HIRSCHGRABEN 9—11 · 5100 AACHEN
TEL. 02 41/2 52 26

APPLE-kompatible Rechner	ab **DM**	**798,—**
IBM-kompatible Rechner mit Laufwerk	ab **DM**	**2198,—**
MC-68000-Rechner als Bausatz oder fertig	ab **DM**	**498,—**

und natürlich alle Erweiterungen wie:

Interface-Karten	ab **DM**	**99,—**
Winchester ab 5 MB	ab **DM**	**698,—**
Floppy-Laufwerke	ab **DM**	**398,—**
Monitore	ab **DM**	**278,—**
Matrix-Drucker	ab **DM**	**698,—**
Typenraddrucker Juki	ab **DM**	**748,—**

Und Service in der eigenen Werkstatt!

Fordern Sie für spezielle Produkte unsere Preisliste an.

ccp datentechnik

TANDON PC 2 Drives, 256 K .	**4 286,—**
PCX 10 1 Drive, 10 MB	**5 768,—**
PCX 20, 1 Drive, 20 MB	**6 235,—**
PCA AT-komp., 2 Dr., 512 K ..	**8 575,—**
PCA-20 1 Drive, 20 MB	**9 580,—**
PCA-30 1 Drive, 30 MB	**11 085,—**
MULTITECH Popular 256 K .	**2 798,—**
mit 2 Drives	**3 197,—**

Zusatzprodukte für IBM auf Anfrage

Zubehör für Apple II/e/c

80 Zeichen + 64 K	**138,—**
640-K-Drives f. Apple //c	**1 090,—**
640-K-Drives f. Apple //e	**798,—**
Doppelstation 2 x 640 K	**1 289,—**
Festplatte 10 MB f. Apple ..	**3 380,—**
Festplatte 20 MB f. Apple ..	**3 985,—**
Mouse für Game-Port	**198,—**

Gratiskatalog anfordern bei:
ccp datentechnik
Herderstr. 12 — 2000 Hamburg 76 — Tel. 0 40/22 56 76

Brückmann/Englisch/Gerits
ATARI ST intern

Düsseldorf, 1985
Data Becker GmbH
306 Seiten
DM 69,-
ISBN 3-89011-119-X

Jeder Besitzer eines Atari 520 ST wird über die mitgelieferte Dokumentation enttäuscht sein. Es war nur eine Frage der Zeit, bis die Fachbuchverlage dem Übel abhelfen würden. Diesem Buch kommt es zugute, daß es nicht zu der Generation Bücher gehört, die schon vor der Markteinführung des Rechners erschienen sind. Die Autoren stellen allerdings einige Ansprüche an das Vorwissen des Lesers, wenden sich also nicht an Computeranfänger.

Am Anfang steht eine Beschreibung der im Rechner verwendeten ICs. Sie ist zwar kurz und knapp, läßt aber die wesentlichen Zusammenhänge erkennen. Unter dem Oberbegriff 'Schnittstellen' beschäftigen sich die Verfasser auch mit der intelligenten Tastatur und ihrer Programmierung. Im dritten Teil des Buches geht es um die Prozeduraufrufe von GEM-DOS, die detailliert erläutert und durch Beispiele gut verständlich gemacht werden. Das BIOS und die erweiterten BIOS-Funktionen sind jeweils mit einem Beispiel in 'C' und Assembler erklärt. Es folgen Erläuterungen der Grafik, der Interrupts und der System-Variablen. Das letzte Kapitel behandelt den Befehlssatz des 68000, der aber für den Anfänger zu knapp beschrieben wird. Der Anhang umfaßt 150 Seiten BIOS-Listing, das größtenteils deutsch dokumentiert ist.

Für denjenigen, der mehr mit seinem Atari machen will, als gekaufte Programme laufen zu lassen, ist das Buch uneingeschränkt zu empfehlen. Ein kleiner Wermutstropfen ist allerdings die schlechte Verarbeitung des Buches, das sich sehr bald in eine Loseblattsammlung verwandelt.

KK

Herwig Feichtinger
Arbeitsbuch Mikrocomputer

München, 1985
Franzis Verlag
602 Seiten
DM 108,-
ISBN 3-7723-8021-2

Es ist ja nicht billig, das 'Arbeitsbuch Mikrocomputer', aber dafür steht viel drin. Es ist vollgestopft mit Informationen, denn der Autor wollte etwas schaffen, '. . .wo man im Zweifelsfall auch mal nachschlagen kann. . .', und zwar 'was man in der täglichen Praxis am meisten braucht'.

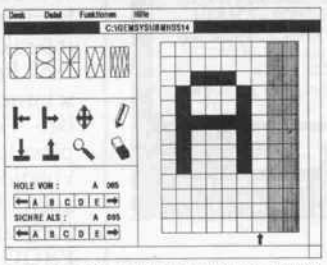
Am meisten braucht der Verfasser offensichtlich 'Hardware eines Computers' (Kapitel 1), bestehend aus CPUs, Speichern, I/O-Bausteinen, TTLs und vielem mehr. Zu allen Themen gibt er eine verständliche, korrekte Einführung oder Erklärung, und dann kommt das, was den Wert dieses Buches als Nachschlagewerk ausmacht: reichlich Tabellen und Abbildungen. So findet man zum Beispiel die Anschlußbelegungen von 21 Mikroprozessoren, von SC/MP bis 80286. Natürlich sind hier nicht alle Bauteile vertreten, die je in einem Computer steckten, aber über die wichtigen hat der Autor die richtigen

Die neue GEM-Software

CCP-Fontedit

Der Fonteditor zu GEM. Erstellen Sie neue Zeichensätze für GEM Draw, GEM Paint und alle anderen GEM-Applikationen, oder benutzen Sie die bereits mitgelieferten neuen Zeichensätze (Pica, Outline und Rustica).

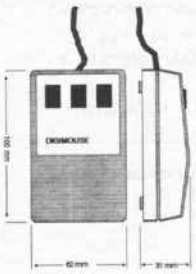
Lieferbar für IBM PC/XT/AT, Siemens, Philips: YES und alle Kompatiblen. Lieferumfang: ausführliche Programmbeschreibung im stabilen Kunststoffschuber, 1 Programm- und 3 Zeichensatzdisketten. **DM 249,—**



TURBO GEM-Tools

Das Graphikpaket zu Turbo-Pascal: Über 80 Routinen unterstützen einfache graphische Bauelemente (Kreise, Ellipsen, Kreisbogen, Rechtecke, Linien, Polygone) und Präsentationsdarstellungen (Säulen-, Kreis- und 3-D-Graphiken). Fenstertechnik und volle Mausunterstützung! Diverse Programmier-Utilities (Generierung von PASCAL-Programmen aus GEM-Draw-Bildern, Programmgenerator für Graphikprogramme).
Graphikausgabe in voller Auflösung und ohne Source-Code Änderung auf diverse Drucker und Plotter! Bilder können als Metafile gespeichert und mit GEM-Draw oder GEM-Paint weiterverarbeitet werden!
Lieferbar für IBM PC/XT/AT (Hercules-, EGA- oder IBM-Farbkarte), Siemens PC-D, Philips: YES und alle Kompatiblen. Lieferumfang: ausführliche Programmbeschreibung im stabilen Kunststoffschuber, 1 Diskette mit zahlreichen Programmbeispielen. **DM 239,—**

Die CCP-Maus



programmierbar auf 6 verschiedene Mausformate (z. B. Microsoft, Mouse Systems oder Summagraphics). Problemlos einzusetzen mit GEM-Software und allen anderen Programmen, die eines der genannten Mausformate unterstützen.
Die CCP-Maus wird standardmäßig als „serielle Microsoftmaus" geliefert und kann an jede serielle Schnittstelle angeschlossen werden. Lieferumfang: deutsches Handbuch und Treiberdiskette. **Preis: DM 559,—**

und noch mehr von GEM:

GEM-Software

GEM Desktop 2.0, deutsch	DM 179,—
CCP-Maus incl. GEM-Desktop 2.0	DM 649,—
Turbo GEM-Tools und GEM Desktop 2.0	DM 299,—
GEM Draw	DM 549,—
GEM Collection (Paint + Write)	DM 549,—
GEM Wordchart	DM 549,—
GEM Graph	DM 769,—
Busy Charts II	DM 449,—

In Kürze lieferbar: ABC Ledgers, GEM-Open Access und BUSY CALC.

Informationen und Beratung zu allen Produkten erhalten Sie über:

CCP Soft
Wilhelmstraße 43, 3550 Marburg/Lahn
Telefon 06421/12104

Der Programmgenerator

TURBO GEN 1

generiert komplette Datenerfassungsprogramme in Turbo Pascal

DM 224,— incl. Source Code

zusätzlich benötigt werden:
Turbo Screen, Turbo Screen +, Turbo Data, Turbo Toolbox

Turbo Pascal und Turbo Toolbox sind eingetr. Warenzeichen von Borland International

Die TURBO Familie

Die weiteren Familienmitglieder

TURBO SCREEN
die Dialog-Toolbox

TURBO SCREEN +
der Maskengenerator

TURBO DATA
der Dateizugriffsgenerator

Programme im Quellcode incl. Handbuch je **DM 224,—**

Handbücher zum Kennenlernen je **19,80**/f. Turbo Screen **39,80**

Zippelhaus 4
2 Hamburg 11

Telefon: 040/337478

CID
COMPUTER-INFORMATIONS-DIENST GMBH

ATARI 260ST/520ST+ ..... günstig! **auf Anfr.**
Floppy 3,5" 1MB ..... **598,—** Doppelfloppy ..... **998,—**
COMMODORE 128 ... **798,—** 128 D ..... **1698,—**
MSX Philips / CE-TEC Zubehör, Floppy's, Drucker ..... **günstig**

Drucker Seikosha
GP 500 ... **373,—** GP 550 ... **485,—** GP 700 ... **825,—**
SP 800 .... **795,—** SP 1000 ... **825,—** MS 15 Typenr. **748,—**
NEC P2 180 Z/s Centronics oder RS 232 ..... **1690,—**

TEAC Laufwerke
FD 55 BV .. **369,—** FD 55 FV .. **395,—** FD 35 F .. **445,—**
SD 510 10MB Festpl. **1448,—** SD 520 20MB Festpl. **1874,—**
20MB Tapestreamer Sliml. incl. Controller f. IBM ..... **2886,—**

PC IBM-komp. 256KB, dt. Tastatur, 1 Laufwerk ..... **1925,—**
NEC 20MB Festplatte incl. Controller WD 1002S ..... **1998,—**

Dipl.-Ing. Jürgen Rost · Computer · Elektronik
Burgstraße 6 3257 Springe 1 Tel. (0 50 41) 46 09

PS COMPUTER VERTRIEB
GbR JÜRGEN POHLSCHEIDT
HENRI SIEBERTZ
Telefon 02 11/72 11 28
Ellerstr. 187
4000 Düsseldorf 1

Sind Sie Hard-/Softwarehändler, Großabnehmer oder Entwickler?
Dann sind wir die richtige Adresse für Sie.

Wir bieten Ihnen die Möglichkeit, mit wenig Kapitalaufwand ein reichhaltiges Sortiment mit guten Einkaufspreisen zu erwerben und das auch schon bei kleinen Bestellmengen.

Harddisk, Laufwerke, Monitore, Cards, usw.

Fordern Sie unverbindlich unsere Preisliste an.
Sie werden erstaunt sein.

VERTRIEB IN GANZ EUROPA

CP/M-Corechner für IBM PC/XT/AT

Der CP/M-Corechner läßt den Betrieb von CP/M und CP/M-80 auf dem IBM und seinen Kompatiblen zu. Die neuen Möglichkeiten, die sich hier bieten sind enorm.

Was bietet der CP/M-Corechner?

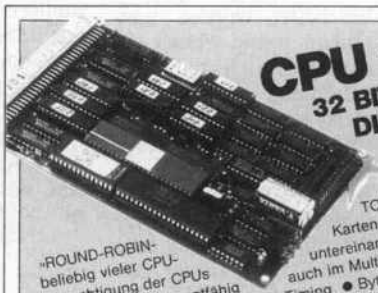
2 x RS 232
1 x Centronic
1 x Battery-gepufferte Echtzeituhr für CP/M und PC-DOS
Optimal: SASI-Schnittstelle IEC-Schnittstelle

CPU: MD 64180 mit 12 MHz-Clock das entspricht einer Z 80 mit 6,6 MHz.
512 KB RAM unter CP/M
64 KB frei für CP/M
320 KB RAM-Disk
120 KB-Drucker-Buffer

256 KB-RAM-Erweiterung für PC-DOS. Austausch von CP/M- und PC-DOS-Files möglich. Damit sie die Daten von Ihrem CP/M-System laden können, liefern wir die nötige Transfersoftware gleich mit.

HÄNDLERANFRAGEN ERWÜNSCHT

FRANK SYPPLIE
Elektronische Baugruppen und Layout
Lindenstraße 24, 7550 Rastatt 1, Telefon (07222) 25662



CPU 150
32 BIT POWER
DIE 2. GENERATION

- 68008 CPU, max. 12,5 MHZ
- multiprozessor-fähig durch TOKEN- ARBITER, Verwendung Karten auf einem Bus, absolute Gleichberechtigung der CPUs untereinander (kein MASTER/SLAVE-Verhalten) - DMA- und interruptfähig - vom 68008-Takt unabhängiges 4 MHz-Z 80-ECB-Timing - Byte-wide-Sockel für 16 KByte EPROM, auch im Multiprozessor-Betrieb - 256 KByte dyn. RAM, integrierter Refresh - 2 x V.24 mit 18 progr. Baudraten - 16 Bit Counter/Timer - Batteriegepufferte Echtzeituhr mit Alarm - bidirektionaler Reset

"ROUND-ROBIN"-beliebig vieler CPU-

ECB-Baugruppe CPU 150:
excl. MWSt DM 1390,-
incl. MWSt DM 1584,60

Telefon 02534/809-0

SYSTEC
Mikroprozessor GmbH · Nottulner Landweg 104 · 4400 Münster
Postf. 410126 · Telex 891551sysd

EPROM Programmier-Platine
anschließbar an jeden Computer mit Centronics-Schnittstelle.

Durch unkomplizierte Schaltung leicht zu handhaben.
Für **Einsteiger** und **Hobbyisten** entwickelt.

Programmiert 2764, 2764A, 27128, 27128A.

Platine mit Handbuch ..... **120,— DM**

Bausatz:
Version A (ohne Netzteil) ..... **128,— DM**
Version B (mit Stromversorgung = 5 V, = 21 V) ..... **149,— DM**

Gegen Aufpreis:
1. Transformator 220/24 V, 4,5 VA ..... **11,20 DM**
2. Centronics-Buchse ..... **8,95 DM**
3. Für geprüftes Fertiggerät ..... **21,— DM**

Das mitgelieferte **Handbuch** beschreibt genau die Schaltung und Software. **Es wird kein Computer-Chinesisch benutzt.**

Lieferung per Nachnahme + Porto. Die Angebote sind freibleibend.

Andreas Sprengel, Entwicklung für Computer- und Steuerungs-Technik
Hinrichsring 15A, 3000 Hannover 1, Tel. (0511) 69 27 70

STAR Drucker
SG-10 80St 120 Z/s **835,—**
SD-10 80St 160 Z/s **1132,—**
SR-10 80St 200 Z/s **1538,—**
SG-15 136St 120 Z/s **1174,—**
SD-15 136St 160 Z/s **1498,—**
SR-15 136St 200 Z/s **1862,—**
Star Drucker sind umschaltbar von IBM-PC Modus auf Centronics parallel. Option RS 232 C

Panasonic Drucker
KX-P 1080 100 Z/s **755,—**
KX-P 1091 120 Z/s **845,—**
KX-P 1092 180 Z/s **1154,—**

Epson Drucker
LX-80 m. Trakt 100 Z/s **898,—**
FX-85 80St 160 Z/s **1297,—**
FX-105 136St 160 Z/s **1725,—**

Typenraddrucker
Olivetti **a. A.**

Schneider Computer
CPC 464 m. gr. Monitor **754,—**
CPC 6128 m. gr. Monitor **1418,—**
Zenith Computer **a. A.**
Farbbänder Nachfüllpack **14,50**
für Star SD/SR, NEC-P2/P3,
Farbbänder für SG-10/15 **10,50**

Vertragshändler ZENITH data systems Die 100% Computer

COMPUTER BÜROMASCHINEN SERVICE
Stormstraße 18, 4430 Steinfurt, Tel. (0 25 51) 25 55
alle Preise inkl. MwSt. zuzügl. Versandselbstkosten
Versand per Nachnahme. Auf alle Geräte 6 Monate Garantie!

c't-Einzelheft-Bestellung

c't können Sie direkt beim Verlag zum Einzelheft-Preis von DM 6,50 (Jahrgang '85 DM 6,—) (zuzügl. Gebühr für Porto und Verpackung) nachbestellen. Bitte fügen Sie Ihrer Bestellung einen Verrechnungsscheck über den entsprechenden Betrag bei.

Die Ausgaben 12/83 bis 1/85, 3/85 und 4/85 sind bereits vergriffen.

Gebühr für Porto und Verpackung: 1 Heft DM 2,—; 2 bis 6 Hefte DM 3,—; ab 7 Hefte DM 5,—.

Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61

Konto-Nr.: 9305-308, Postscheckamt Hannover

Konto-Nr.: 000-019968 Kreissparkasse Hannover (BLZ 250 502 99)

Informationen zusammengetragen.

Nicht so dringend werden Peripheriegeräte gebraucht. Das zweite Kapitel umfaßt nur 50 Seiten und geht kaum ins Detail. Zum Beispiel sind zwar Steuer-Codes für Matrixdrucker angegeben, aber nur für den Star DP-8480 und nicht für die viel verbreiteteren Epsons (etwa FX-80). Über Terminals wird gar nichts gesagt.

An dritter Stelle steht die 'Programmierung von Mikrocomputern', gefolgt von 'Mikrocomputer-Betriebssysteme' und 'Anwender-Software'.
Der Schwerpunkt liegt in diesen Kapiteln auf Apple II und CP/M-80. Auf den C 64 und die Betriebssysteme MS-DOS und Unix wird aber auch eingegangen. Außerdem finden sich in diesen Kapiteln wichtige Zusammenstellungen: Befehle verschiedener Versionen von BASIC, Pascal und Forth, Betriebssystemkommandos sowie Kommandos von WordStar und Applewriter-II.

Den Schluß bilden eine 10seitige Adressensammlung (kann sehr nützlich sein!) und ein ausführliches Stichwortverzeichnis.

Das Buch ist sorgfältig bearbeitet worden und scheint wenig Fehler zu enthalten (ich habe 11 gefunden), die zudem meist leicht zu erkennen sind ('... bekannteste Fortran-Compiler ist BASIC-80'). Der einzige schwerwiegende Fehler, der mir auffiel, war eine falsche Beschaltung des 75188 auf Seite 115. Die BASIC-Beispiele sind leider schwer zu lesen, denn auf Leerzeichen hat der Verfasser konsequent verzichtet.
Wie man sieht, gliedert sich das Buch grob in einen Hardware- und einen Software-Teil. Ersterer ist ausführlich und universell anwendbar. Der Software-Teil ist dagegen zu sehr auf Apple II und CP/M spezialisiert, bietet auf diesen Gebieten dann aber auch viel Information.

Für Besitzer von Apple- und CP/M-Computern, die sich nicht nur mit ihrer eigenen Maschine, sondern auch mal mit den Ideen und Computern anderer Leute beschäftigen, ist dieses Buch genau richtig. Dann kann es tatsächlich zum Arbeitsbuch werden, das sein Geld wert ist. AU

John Bear
Computerfrust – ein Vermeidungshelfer

Reinbek, 1985
Rowohlt
Taschenbuch Verlag
286 Seiten
DM 14,80
ISBN 3-499-18125-8

In einer Zeit grassierender Computerlust – ein Buch über Computerfrust? Soll angehenden Computerfans ausgeredet werden, sich auf die neue Technik einzulassen?

Im Gegenteil: 'Lust statt Frust' heißt das Motto. Der Autor, selbst leidgeprüfter Autodidakt, möchte anderen teures Lehrgeld ersparen. Er vermittelt seinen Lesern das nötige Grundwissen und eine Vielzahl praktischer Tips, um beim Kauf eines Computers die bestmögliche, auf die individuellen Bedürfnisse zugeschnittene Wahl treffen zu können. Das Buch wendet sich also an absolute Neulinge.

Der Verfasser kommt ohne jedes Fachchinesisch aus und legt seine Erfahrungen im Umgang mit Computern in einer schnoddrig-griffigen Sprache vor, die die Lektüre zugleich informativ und amüsant macht. So können wohl nur Amerikaner schreiben.

Allerdings verführt ihn

NEU NEU

B & S-Schallschluckhauben — die deutsche Alternative

— Jeder 2. Drucker liegt über dem zugelassenen Lärmpegel von 55 dBA. — Tun Sie was dagegen, damit Ihre Mitarbeiter motiviert und unter normalen Bürobedingungen arbeiten können.

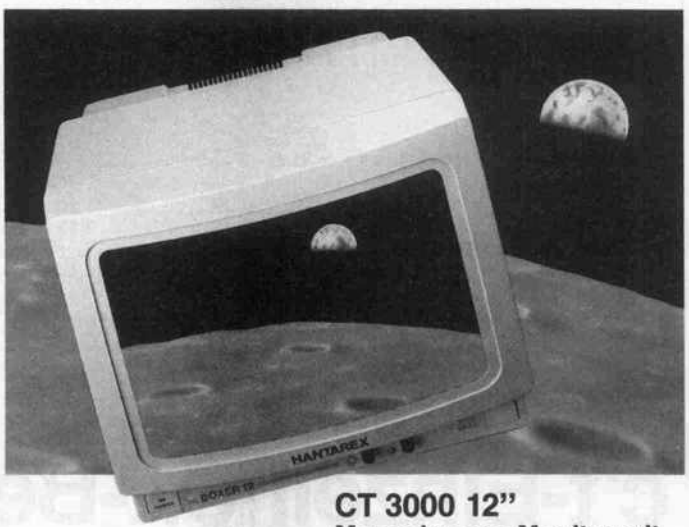
— B & S-Hauben vermindern den Lärmpegel von 70—90 % je nach Typ.

— Wir bieten Ihnen: „Kostenlose Probestellung von 1 Woche zum Kennenlernen".

— Nutzen Sie diese Möglichkeit. Sprechen Sie mit uns!

B & S — Bohne & Speidel GmbH
Hohenzollernstr. 77, D-7530 Pforzheim
Telefon 07231/102281

... sprengt alle Dimensionen



CT 3000 12"
Monochromer Monitor mit entspiegelter Bildröhre. Optimale Streulichtunterdrückung durch dunkles Glas. 20 MHz Video-Bandbreite. Abbildungsformat 80 x 25 Zeichen. Auch mit NF-Verstärker und Lautsprecher lieferbar.

HANTAREX
Deutschland Vertriebsgesellschaft mbH

Siegener Straße 23
5230 Altenkirchen
Tel.: 0 26 81/30 41/42
Telex: 869 991 hantx d

SONIC 300

Akustikkoppler
FTZ 18.13.1967.00

SONIC GmbH
Paul-Marien-Straße 23
6600 Saarbrücken
Tel.: 0681-63 51 31

AUSTRIA
MICRO STUDIO
HOFMÜHLGASSE 1
1060 WIEN
Tel.: 0222-5668 204

Technische Daten SONIC 300:

CCIT-Norm V21 / V 24 (RS 232 C): 300 Baud Vollduplex, TD, RD, DTR, DSR, RTS, CTS, DCD (als DCE gesch.)

Netzanschluß 220 V AC (integriert im Koppler), Batteriebetrieb 2x9 V, Akkubetrieb 2x9 V Akkus werden bei Netzbetrieb automatisch aufgeladen.

originate, answer, Selbsttest (analog loopback), Echosperrenunterdrückung, stand by Betrieb

7 LED-Anzeigen für originate-answer-EIN/AUS-ready, selftest-stand-by-level

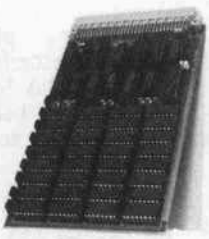
moderne Tastatur (Kurzhubtaster), speziell entwickelte Gummimuffen für hohe Übertragungssicherheit.
Gerät ist vom Rechner aus einschaltbar.

Preise zuzüglich Versandkosten:

SONIC 300	298,– DM
V24-Schnittstelle für C64/VC20	78,– DM
Kabel 25 pol.	38,– DM
SONICOM Terminalprogramm auf Disk.	20,– DM
64er Paket (SONIC 300, Schnittstelle, Kabel, SONICOM)	398,– DM

NEU!
SONIC 300CL mit FTZ 198,–

1 MB-ECB-Ram-Disk



- kompatibel zu jedem ECB-Bus
- einsatzfähig am Z80, 8086, HD64180
- keine Waitzyklen
- Installationen für CP/M 2.2, CP/M 3.0 und MDOS
- Teilbestückung in 256-KByte-Schritten
- alle ICs gesockelt

— Fertiggerät getestet . . . . . **855,—**
— Bausatz komplett (1 MByte) . . **529,—**

Chr. Kayser
Ludwigstr. 29, 3300 Braunschweig
Tel.: 05 31/34 15 32

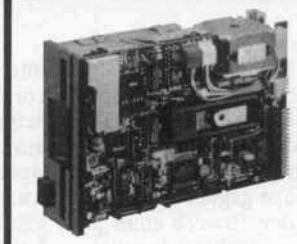
TIPS P50 zum Thema Terminal

Low-Cost Video-Subsystem für Z80/ECB-Bus

Mit Bildformat 80x25 Zeichen und der implementierten TVI950 Emulation stellen alle Funktionen zum Betrieb von Standardsoftware zur Verfügung. TIPS P50 verfügt über ein Bildattribut für inverse Schrift oder Anwahl eines zweiten Zeichensatzes. Die Synchronisationslogik ermöglicht der TIPS-eigenen CPU und dem CRT-Controller abwechselnden Bildzugriff. Dies gewährleistet sehr schnellen und bildstörungsfreien Betrieb. Tastaturanschluß parallel oder seriell und Funktionstastenumsetzung (Stringausgabe) möglich. Piezo-Schallgeber ermöglicht Keyklick u. BEL. TIPS-Videokarten bieten Industriequalität: Probimer-Platine, maschinelle Fertigung, eine Versorgungsspannung und Überspannungsschutz.

TIPS P50 komplett: DM 533,52 (DM 468,- + MwSt.)
Infos zu TIPS P50, TIPS 100 und TIPS S50 kostenlos.

ELSA — Gesellschaft für elektronische Systeme mbH
Monheimsallee 53 · D-5100 Aachen · Tel. 02 41 / 2 99 92 · Tx 8 329 791



Epson FDD 3,5" und 5,25"
Beide Laufwerkstypen sind von höchster Qualität und extrem preisgünstig. – Die FDD 3,5" sind **batteriebetreibbar,** also unabhängig von der Steckdose! Dadurch sind **mobile Datenspeicher** konzipierbar. 3,5"-Laufwerke der 100er-Serie: bis zu 1 MegaByte.
5,25"-Laufwerke der SD 500-Serie: bis zu 1,6 MB Speicherkapazität.

Mehr Informationen:
Telefon (06151) 313890 und 312693

EBG — Elektronik-Bauelemente · Computer-Systeme · Darmstadt

EBG Darmstadt
Elektronik-Bauelemente + Computer-Systeme
Lothar Schanuel GmbH
Heidelberger Str. 73, 6100 Darmstadt
Tel. (06151) 313890/312693
Telex 4197160 shan d

Unglaublich

Lötstation Thermotronic 50

— elektronisch geregelt
— mit Ablageständer

DM 128,—

Sonderliste 11/85 anfordern.

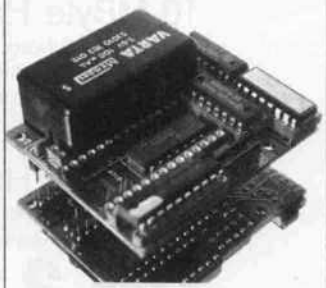
HANSA ELECTRONIC GMBH

Schopenhauerstr. 2
Postfach 546
D-2940 Wilhelmshaven
Telefon 04421/38773

REED
DATENTECHNIK

Systemhaus + EDV-Beratung
Lütge Bruch 1a · 4690 Herne 1
☎ *02323/33163*

NEU ct Uhr



Die Echtzeituhr für jeden Rechner

Mit Akku & Schaltjahreskorrektur. Einfach einem EPROM unter die Beine schieben — fertig. Software für viele Rechner verfügbar, z. B. für **IBM-PC, ATARI 520 ST, C64, CP/M (Schneider CPC),** . . .

Preise incl. MwSt.:

Leerkarte mit PAL	53,—
kompletter Bauteilesatz dazu	58,—
fertig aufgebaut und getestet (bitte Rechnertyp angeben)	169,—
für enge Platzverhältnisse mit Flachbandkabel	174,—
IBM Dev. Treiber auf Disk (Uhr wird Teil des DOS)	15,—

Dipl.-Ing. Eberhard Meyer
Computer & Elektronik
Keplerstr. 6, 3000 Hannover 1
Tel. 0511/3504500

Ehrensache, . . .

daß wir Beiträge und Bauanleitungen aus inzwischen vergriffenen c't-Ausgaben für Sie **fotokopieren.**

Wir müssen jedoch eine Gebühr von **DM 5,— je abgelichteten Beitrag** erheben — ganz gleich wie lang der Artikel ist. Legen Sie der Bestellung den Betrag bitte **in Briefmarken** oder in Form eines Verrechnungsschecks bei. **Und: bitte, Ihren Absender nicht vergessen.**

Folgende c't-Ausgaben sind vergriffen:
12/83 bis 1/85, 3/85 und 4/85.

c't magazin für computertechnik, Verlag Heinz Heise GmbH, Postfach 610407, 3000 Hannover 61

die Lust am Fabulieren streckenweise zu einer gewissen Banalität. Da werden für den Computerkauf allgemeine Ratschläge gegeben, die sich für den Erwerb eines jeden Gebrauchsgegenstandes, vom Stuhl bis zum Schnellkochtopf, von selbst verstehen. Über solche redundanten Passagen hilft die humoristische Qualität des Buches jedoch unterhaltsam hinweg.

Die weitergehenden Informationen und das technische Grundwissen, das man zum Kauf eines Computers besitzen sollte, wurden praxisnah und 'problembewußt' ausgewählt und können eine echte Hilfe darstellen.

Fazit: Das Buch ist bestens geeignet, ins kleine Einmaleins der Digitaltechnik einzuführen und kauflustigen Computerinteressenten die Angst vor der neuen Technik zu nehmen. Aufgrund seiner umwerfenden Komik eignet sich 'Computerfrust' aber auch für Fortgeschrittene – als launige Lektüre zum Thema 'Der Computer und ich'. MF

Sgonia/Warner
Turbo Pascal

Tips & Tricks
Düsseldorf, 1985
Data Becker GmbH
243 Seiten
DM 49,-
ISBN 3-89011-091-6

'Tips & Tricks', das klingt so, als würde man in die Besonderheiten und tiefsten Geheimnisse von Turbo-Pascal eingeweiht. Der Inhalt dieses Buches ist aber weit weniger speziell – und damit viel nützlicher. Das Buch nämlich enthält erläuterte Programme und Programm-Bausteine, die man in eigene Programme integrieren kann.

Auf den ersten 86 Seiten geht es ums Sortieren und um Bäume, also Dinge, die man immer wieder braucht. Die verwende-

ten Verfahren werden gut erklärt, und die Beispielprogramme sind nachvollziehbar programmiert. Insbesondere wird hier auch das Sortieren auf Diskette behandelt, worüber man sonst wenig liest.

Im zweiten Teil des Buches geht es um eine bessere Zusammenarbeit zwischen Turbo-Pascal und dem Computer, speziell um 'erweiterte Bildschirmausgabe-Routinen', 'komfortable Eingabefunktionen', 'Zeit und Datum' und einiges mehr. Dieser Teil ist hauptsächlich auf MS-DOS abgestimmt, aber wenn sich die Programme auf CP/M-80 übertragen ließen, wurde dies getan. In solchen Fällen sind zum Teil beide Versionen abgedruckt. Zum Verständnis dieser Programme braucht man einiges Wissen über das Betriebssystem – die Autoren geben Hinweise auf einschlägige Literatur.

Den Schluß bilden drei längere Listings eines Programmlisters, eines Crossreferenz-Erzeugers und eines Tracers, die einschließlich Erklärungen 50 Seiten füllen. Ob man diese Programme braucht und eintippen mag, muß jeder selbst entscheiden.

Das Buch enthält eine Menge Informationen und viele nützliche Programm-Bausteine. Ich fand den ersten Teil am interessantesten, denn er ist so allgemein geschrieben, daß man auch als Benutzer eines anderen Pascal-Compilers (vielleicht sogar einer anderen Programmiersprache) etwas davon hat. Auch die anderen Teile sind so gut, daß sie fast jeder für sich den Kauf des Buches rechtfertigen. AU

RATEV Aktuell
Peripherie für IBM* und kompatible Systeme

10 MByte Harddisk Paket
Winchesterlaufwerk (jap. Markenfabrikat)
Western Digital Controller WD 1002 S WX 2
DM 1 745,—

20 MByte Harddisk Paket
Winchesterlaufwerk (jap. Markenfabrikat)
Western Digital Controller WD 1002 S WX 2
DM 2 150,—
Einzelkomponenten auf Anfrage

EPSON Floppy Laufwerke

3,5" SMD 150, 250 KB, 1x40 Spur	DM 350,—	5,25" SD 521, 500 KB, 2x40 Spur	DM 355,—
3,5" SMD 160, 500 KB, 2x40 Spur	DM 440,—	5,25" SD 540, 1 MB, 2x80 Spur	DM 385,—
3,5" SMD 170, 500 KB, 1x80 Spur	DM 350,—	5,25" SD 560, 1,6 MB, 2x77 Spur	DM 435,—
3,5" SMD 180, 1 MB, 2x80 Spur	DM 420,—	5,25" SD 580, umschaltbar 77/80 Spur	DM 525,—

Riteman Drucker
Riteman C+, der Drucker für den C64 DM 849,—
Riteman F+, der EPSON FX-80 kompatible DM 945,—

* IBM Warenzeichen der International Business Mashines Corp. alle Preise inkl. 14% MwSt.

RATEV Elektronik Vertriebs GmbH
Postfach 16 01, Gothaerstr. 15, 4030 Ratingen 1
Tel.: 0 21 02-4 20 51-52 Telex: 85 85 180 Mailbox: 0 21 02-47 54 00
Besuchen Sie auch unser Ladenlokal!

APPLE + IBM kompatibles

Info geg. DM 1,40 in Briefmarken

SPRINGMANN COMPUTER GmbH
Aegidientorplatz 2A
D-3000 Hannover 1
Tel: 0511-880080 Tlx: 921466 comps d

PC-48 Computer, 48 KB, o.EPROMs 695,-
PC-64 Computer, 64 KB, o.EPROMs 795,-
PC-64xE Computer, 64 KB + 256, o.EPROMs 1095,-
IBM-look, abgesetzte Tastatur

Palete bestehend aus Computer,Disk, Controller:
Paket 48, incl. 1 DISTAR + Controller 895,-
Paket 64, incl. 1 DISTAR + Controller 995,-
Paket 64xE, incl. 2 DISTAR + Controller 1795,-

Zubehör / Zusatzkarten für Apple II:
Unser umfangreiches Angebot über APPLE-kompat. Produkte finden Sie in unserer INFO.

Tastaturen:
Keyboard,CHERRY-Tasten, deutsches Layout, 255,-
10programmierbare Funktionstasten,für Apple II und -kompatible, m.3015 Stecker
Keyboard,CHERRY-Tasten f.IBM-PC, US ASCII 255,-
Keyboard dto., jedoch deutsches Layout 255,-

16-Bit Computer und Zubehör:
PC-8237 Komplett-Computer, IBM-PC-komp. 2192,-
256KB bestückt, Netzt, Gehäuse, Keybd, Multi-I/o-Drivecard, Color Graftic Card, 360KB-Disk
PC-8817 Motherboard, IBM-PC compatibel, 619,-
0 KB bestückt, bis 640 KB erweiterbar,
8 Slots, Disk-Contr, 2ser, 1par. on board
PC-8237 Motherboard, IBM-PC compatibel, 416,-
0 KB bestückt, bis 640 KB erweiterbar,
PC-8787.T Motherboard, IBM-PC kompatib. 498,-
Spezifik. wie PC-8787, jedoch 8MHz Taktfrequenz
Netzteil f.PC-87 / PC-88 269,-
Color Graphic Card 205,-
Monochrom Graphic/Printer Card 299,-
RAM-Card 1, 6 Stck. 28pol. Sockel 59,-
512K Karte, OKB bestückt 169,-

Bladelay-Preise per Zeile:
Ab 6 Zeilen und mehreren Anzeigen der Anzeige liegen 5 Wochen. Bewusste Preise erfahren Sie telefonisch.

Multifunction Card MF 605 max.256KB, OKB bestückt, 1par. 1ser. Interf., 2.ser.möql, Uhr mit Akkupufferung 219,-
Multifunction Card MF 610 m.o, jedoch max.384KB, nur 1ser.Intfc.möql, dafür mit Joystick-Anschluß 349,-
MULTI-i/o /DRIVE Card, Disk-Controller, 1 parall. + 1 seriell.Intfc, 2.ser.möglich, Uhr mit Akkupufferung, Joystickanschluß 369,-
Seriell.Interface, 1 Port bestückt, 2.opt 109,-
Parallel-i/o /Timer Card, 48Bit i/o Uhrenbatte m.Akkupufferung und Software 159,-
AD/DA Card, 12Bit, 16 AD- + 1 DA-Kanal 135,-
Joystick mit Autocenter u.Trimmung 299,-
Joystickadapter 65,-
Hire-Wrap Karte 69,-

Harddisks:
Harddisk-Laufwerk 20MB 5"-slimline 1798,-
Harddisk-Controll. f.IBM-kompat. m.Kabel 529,-
Diskettenlaufwerke:
DISTAR f. Apple II, II+, IIe 369,-
DISTAR f. Apple IIc 389,-
Diskettenlaufwerk 48 tpi, 2-seitig, 5.25", slimline, Shugart-Bus, für IBM-PC 269,-
Monitore:
Color-Monitor TVM MD-3 14" umschaltbar auf monochrom grün/bernsteinweiß oder 16 Farben, antireflex, 0,43mm-Maske, RGB 1498,-
Matrix-Drucker:
Logitec KDC FT-5062 (262/s, 40,80,132/2 Schrift, 999,-
40-Zeich.download, 8 Schriftart.(wählbar) beste NLQ, 11 internat.Z-sätze, Traktor+Friktion

HÄNDLER GESUCHT!

IBM-POWER

PC-XT-kompatibles Standardsystem:
640K RAM, 2 Laufw. mit je 360 k, Centronics-Druckerschnittstelle, eine serielle Schnittstelle (eine weitere durch IC nachrüstbar), Game-Port, akkugepufferte Uhr, Farbgrafikkarte, Tastatur **2995,— DM**

Wir versuchen nicht durch absolute *Kampfpreise* zu überzeugen, sondern durch Qualität.
Bessere Materialien sind auch im Einkauf teurer. Trotzdem bieten wir das PC-XT-Startsystem schon ab 1998,— DM an.

Schmidtke Computertechnik
Sandkaulstr. 41, 5100 Aachen
Tel. 02 41/2 32 17

Geschäftszeiten:
Mo.—Fr. 10.00 bis 13.00 Uhr
und 14.30 bis 18.30 Uhr
Sa. 10.00 bis 14.00 Uhr
donnerstags geschlossen

IBM ist ein eingetragenes Warenzeichen

PC-XT Festplatten-Komplett-System mit Monitor:
Daten wie PC-XT jedoch zusätzlich eine Festplatte 20 MByte mit Controller, Monitor mit 22 MHz, grün, 12", Superpreis **5998,— DM**

Als Laufwerke finden nur Original TANDON- od. QUME-Laufwerke Verwendung. Als Festplatte wird MICRO-SCIENCE eingesetzt.

Der Controller verwaltet bis über 60 MByte und bis zu 2 Festplatten.

basys Bauelemente + Systeme GmbH

ELECTRONIC-VERTRIEB
Postfach 1529, D-8039 Puchheim
Tel. 089/801069, Telex 5213385 maus d

AMPEX-TERMINALS

A 210 – 14"
A 230 – 14"
A 219 – 14"
ANSI-KOMPATIBEL
VT 100 / VT 52 / VT 131-kompatibel*
A 220 – 14"
VT 220-kompatibel*

ERGONOMISCH
ANZEIGE:
AMBER oder GRÜN
SMOOTH-SCROLL
SENSATIONELLER PREIS

* VT 100 / VT 52 / VT 131 und VT 220 sind eingetragene Warenzeichen der Digital Equipment Corporation

CP/M-68K für c't 68000 (in Kürze: c't 68 ECB)

- besonders leistungsfähige und schnelle Implementation s. Artikel in c't 9/85
- RAM Disk, Track Puffer, Speicherausbau anpaßbar, Line Editing, BIOS z. T. im source code, 1 5-1/4" und 8" komp. Laufwerke (1.44 MB)
- ab Lager lieferbar
- seriennahtige Tools wie C-Compiler, Assembler, Debugger
- Weitere Tools (z. B. DU68K, CRTX) und Programmiersprachen (z. B. Pascal, Fortran) lieferbar
- inkl. org. Handbücher DM 995.- inkl. CEDIT 1315,- inkl. CEDIT und DU68K 1395,-

CEDIT

Editor für alle CP/M-68K und OS-9 Systeme

- der komfortable u. leistungsfähige Full-Screen-Editor für alle CP/M-68K und OS-9/68000 Systeme wie **mc 68000, NDR Klein, ELTEC, KWS, c't 68000, c't 68 ECB,...**
- an Bildschirm, Tastatur und Speicherausbau anpaßbar
- besondere Features: settliches Scrolling, umschaltbar 132/80 Zeichen/Zeile, Rechenfunktion, Merker, intelligenter und schneller Bildaufbau, gleichzeitige Bearbeitung von bis zu 5 Texten
- inkl. ausführlichem Handbuch für CP/M-68K **DM 390.-**, OS-9 **DM 590.-**

CP/M-68K ist Warenzeichen von Digital Research Inc.
OS-9 ist Warenzeichen von Microware

HSP GmbH • Nottulner Landweg 104 • 4400 Münster • Telef.: 02534/8454

DIE SUPER - MODEM - KARTEN SIND DA!

Blitzschnelle Montage durch stecken.
300, 600 + 1200 Baud.
FÜR DEN EXPORT! OHNE FTZ
Mit deutschem Handbuch und umfangreicher Software

Für APPLE II, +, e oder VC 64 ..... DM 348.-
Für IBM + kompatible ..... DM 498.-
512 K-Karte für APPLE II, +, e ..... DM 298.-
ErPhi-Controller für Industrie-Laufwerke ..... DM 285.-

Bitte fragen Sie die aktuellen Preise
unserer übrigen APPLE + IBM Zusatzkarten an.

Mohwinkel Elektronik, Tel.: 02149/ 37 81 + 9 50 60
Berliner Str. 73, Postfach 25 01 66
5090 Leverkusen / Fettehenne

IBM-kompatibel

Centronics Interface	89,-
Disk Controller	125,-
Color Graphik-Karte	169,-
Multifunktionskarte 1 x Centr. 1 x RS 232, 1 x Game, 1 x Uhr u. Platz für 384 KB RAM	339,-
512 K-RAM-Karte (0 KB)	179,-
64-KB-RAM-CHIP-Satz z. Aufrüsten v. Multif.- und 512 Karte in 64 K-Schritten nach Ihrem Bedarf.	49,-
Monochrome-Karte	279,-
Disklaufwerk 360 K	298,-

Alle Karten werden mit Anleitung geliefert. 1a Qualität mit vergoldeter Steckleiste, Bauteile 1. Wahl u. mehrfach getestet.

Klaus Jeschke
Hard-, Software
Viertstr. 3—16
6233 Kelkheim
☎ (06198) 7523

Händleranfragen erwünscht.
IBM-Info 1/86 für 1.- Porto.

Telefon (02 41) 3 49 62
Noppiusstraße 19, 5100 Aachen

RÖCKRATH MICROCOMPUTER

Für Prof. mc – CP/M – System, Apple – II mit Softcard
Jetzt ab Lager lieferbar: *Altos, Osborne, Epson,*

CP/M-2.2-Assembler Listing

Schneider CPC 464, 664, 6128
Kaypro, C128, ...

* disassembliertes und ausführlich deutsch kommentiertes Assemblerlisting von BDOS und CCP (auf allen Rechnern identisch)
* mit Labels, daher wie Sourcecode!
* genaue Erklärung aller BDOS – Calls
* Tricks und Tips zur BDOS – Erweiterung

ISBN 3 – 925074 – 11 – 2, über 200 Seiten,
DIN – A5, kartoniert 59,– – DM

c't magazin für computer technik

6/86 — Anzeigenschluß am 8. April 1986

Yazdani/Kandler
Computer für Anfänger

Reinbek, 1985
Rowohlt
Taschenbuch Verlag
188 Seiten
DM 9,80
980-ISBN 3-499-17550-9

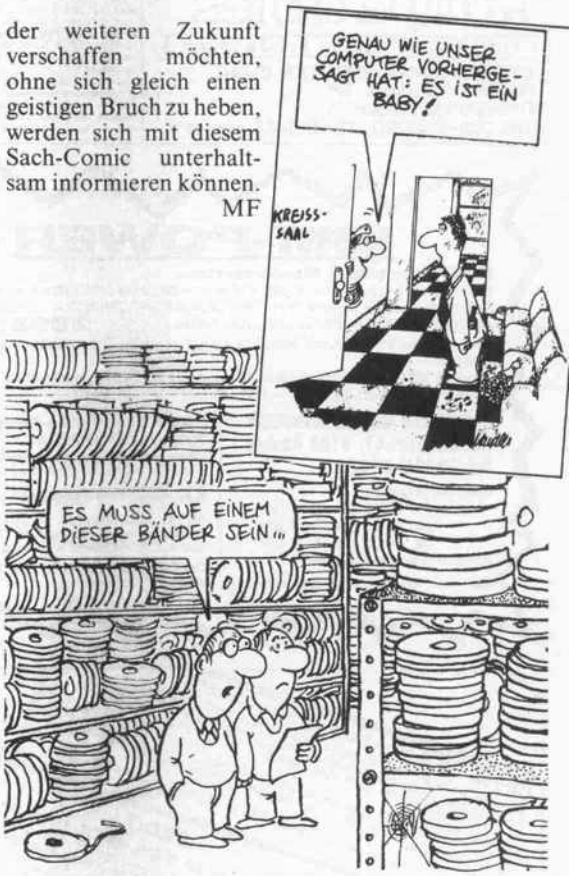
Ein weiterer Sach-Comic aus der 'Anfänger'-Reihe von rororo: Nach Darwin, Marx, Freud und Einstein, Atomkraft, Genetik und anderen Themen nun die Computertechnik, dargestellt in Form von Bildergeschichten und knappen Texten, aufgelockert durch Witzchen und Sprechblasenscherze – geht das überhaupt?

Es geht, und sogar gut. Das Buch ist didaktisch geschickt aufgemacht, liest sich leicht und locker und ist trotz seiner populären Machart keineswegs simpel. Nicht nur werden die technischen Möglichkeiten der Computerei ausführlich behandelt, auch die historischen Hintergründe, die wirtschaftlichen und militärischen Konsequenzen und die Auswirkungen auf das menschliche Selbstverständnis werden kritisch bedacht.

Die wahrhaft 'gründliche' Einführung setzt bei der Schaffung der ersten Handwerkszeuge an, um auf die altertümlichen Denkwerkzeuge zu kommen; leitet von einer anschaulichen Symboltheorie anhand von Alltagsbeispielen auf das binäre System über; erläutert die Funktionsweise von Computern mit Beispielen von Steuerungsprozessen in mechanischen Geräten (wie dem Auto) und führt so auf eine Weise in die schwierige Materie ein, die jeder anhand eigener Erfahrungen nachvollziehen kann. Auch wenn's gegen Ende des Buches komplizierter wird, bereitet die Lektüre keine Mühe – und recht vergnüglich ist sie obendrein.

Natürlich empfiehlt sich das Buch vor allem für die Youngsters unter den Computer-Anfängern, denen diese lebendige Einführung in die Geheimnisse des Computers das Studium trockener Begleitbücher erleichtern könnte. Aber auch 'Späteinsteiger' oder solche, die sich 'nur mal so' einen Einblick in die Technik der weiteren Zukunft verschaffen möchten, ohne sich gleich einen geistigen Bruch zu heben, werden sich mit diesem Sach-Comic unterhaltsam informieren können.

MF



APPLE- und IBM-COMPATIBLE

– mit **6502 + Z80 + 64K RAM + 12K ROM on board**, d. h. 100% Apple-kompatibel und CP/M-fähig mit 2 Zeichensätzen (dt. + ASCII)
– neues Metallgehäuse mit Schaltnetzteil +5 V/5 A
– 2 DISTAR-Laufwerke mit je 163 KByte
– eingebauter Disk-Controller für 2 Drives
– 6 freie SLOTs
– Monacor Daten-Monitor CDM-1200 22 MHz grün oder bernstein
– acs-Tastatur wie Abb. programmierbar mit WordStar-Belegung
– oder Tastatur im IBM-Design mit 10 frei progr. Tasten ASCII- oder deutscher Zeichensatz
– 400seitiges Anwenderhandbuch in deutsch

komplett inkl. MwSt. DM 2 420,—

Apple IIe-kompatible Systeme ab DM 749,—

Die Systemkomponenten können je nach Kundenwunsch zusammengestellt werden (andere Controller, Disk-Drives, Monitore usw.)

Fragen Sie nach einem Angebot, daß auf Ihre speziellen Wünsche zugeschnitten ist!

IBM XT-compatibles Komplett-System

– mit 512 K RAM, 8 freie SLOTs und Boot-ROM
– 1 TEAC FD-55B Laufwerk 2 x 40 Track
– Multi I/O-Card mit Parallel-Port, Seriell-Port, Game-Port, Uhr, Disk-Controller-Anschluß
– Color-Graphik-Karte
– Tastatur (deutsch oder ASCII)
– 135 W Netzteil

komplett inkl. MwSt. DM 2 199,—

TEAC FD-55F
Laufwerk 80 Track ds **DM 399,—**

Super-Preise für PLANTRON PC und AT!!!

Fordern Sie unsere Apple- und IBM-Zubehörlisten gegen DM 2,— in Briefmarken an!

Electronic-Köller
Lothe - Niesetalstraße 4
4938 Schieder-Schwalenberg 4
Telefon 0 52 33/75 50

Ihr GENIE-Stützpunkthändler informiert!

GENIE 16 C
640 kB RAM, 30 % schnellerer CPU (V-Chip), Farbgrafik, 2 LW je 360 kB, diverse Schnittstellen, stabiles Gehäuse und deutsche Tastatur, Betriebssystem MS-DOS, BASIC, TOPTIP **DM 4995,-**

bzw. wie vor, jedoch 1 LW 360 kB, Festplatte 10 MByte **DM 7995,-**
oder
die bewährten 8-bit-Systeme mit dem riesigen Softwareangebot (TRS-80-kompatibel):

GENIE IIs
64 kB, 1 Laufwerk 720 kB, Controller, G-DOS 2.4 **DM 2995,-**

GENIE IIs
64 kB, wie vor, jedoch 2 Laufwerke **DM 3795,-**

sowie eine große Anzahl Erweiterungsplatinen zu GENIE IIs

oder der STERN unter den 8-bit-Rechnern:

GENIE IIIs
128 kB, 7.2 MHz, 2 LW je 720 kB, HIRES-Grafik, diverse Schnittstellen, G-DOS 2.4 und Datenbank DIS. **DM 5900,-**

bzw. wie vor, jedoch mit dem neuen HANTAREX Monitor CT 3000 DM 6250,-

Monitore, Drucker, Verbrauchsmaterial ständig zu günstigen Preisen am Lager.
▶ z. B. 2000 Blatt blanko Druckerpapier 12" × 240 mm DM 46,- portofrei in der BRD.

Star-Farbbänder & Disketten

Wir liefern auch auf Teilzahlung zu günstigen Bedingungen, Anfrage lohnt! Unsere Preise verstehen sich **einschließlich** 14 % MWSt.

Lassen Sie sich unverbindlich informieren. Kontakten SIE uns!

COMPUTER CENTRUM
SOFTWARE · HARDWARE · SERVICE
D-5870 HEMER
HADEMAREPLATZ 60
TEL. 02372/4227

Das große Luther-Standardwerk der 51 Basic-Dialekte

beste Kritiken in Fach- und Wirtschaftszeitungen. Patentfaltung 1375 x 980 mm (1,3475 qm) + zus. 96 Seiten, Buchformat 144 x 278 mm nur DM **49,80**

Verlag Heinz HEISE GmbH · Postfach 61 04 07 · 3000 Hannover 61

Wo immer Sie das BASIC-Listing eines Computers finden — sei es in Zeitschriften, Büchern, Clubmagazinen etc. — **mit dieser Tabelle können Sie alle rechnerspezifischen Sonder- und Graphikbefehle, Ein- und Ausgabebefehle für Bildschirm, Drucker, Kassetten und Disketten, Funktionen und Systembefehle in ihrer konkreten Anwendung nachschlagen.** Bei Konvertierungsarbeiten können Sie sofort den für Ihren Computer zutreffenden Befehl ablesen. Computerumsteiger und Neulinge können mit Hilfe dieser Tabelle den Rechner ausfindig machen, der den von Ihnen benötigten BASIC-Befehlsvorrat hat, so daß die zu lösenden Probleme auch bewältigt werden können. Die große BASIC-Referenztabelle ist auch die große Hilfe im BASIC-Unterricht, da sie **eine bisher nicht dagewesene Vollständigkeit von BASIC-Dialekten im Zusammenhang** bietet.

Zutreffendes bitte ankreuzen:
☐ per Nachnahme
☐ V.-Scheck liegt bei

BESTELL-COUPON

Ja, senden Sie mir „Die große BASIC Referenztabelle der 51 Dialekte" zu DM 49,80 + DM 3,50 Porto u. Verp.

Name/Firma: .....
Straße: .....
PLZ/Ort: .....

Verlag HEISE · Postf. 61 04 07 · 3000 Hannover 61

Heise/ Luther

MAIL-SHOP

Computer-Peripherie
Inh. Georg Zeulner
Alsterdorfer Straße 201
2000 Hamburg 60

Aus unserem Lieferprogramm bieten wir an:

SEIKOSHA-MATRIXDRUCKER
Die flüsternden Alleskönner

SP-800A	CENTRONICS EPSON-FX-80 kompatibel
SP-800I	CENTRONICS IBM-PC Zeichensatz
SP-1000A	CENTRONICS EPSON-FX-80 kompatibel
SP-1000AP	Anschlußfertig für APPLE IIC/MACINTOSH
SP-1000CP	CENTRONICS Anschlußfertig CPC 464
SP-1000VC	Anschlußfertig für Commodore alle Typen
SP-1000MXC	Anschlußfertig für MSX-Computer
SP-1000I	CENTRONICS IBM-PC Zeichensatz
SP-1000AS	RS-232C (V. 24)

Ältere SEIKOSHA-Modelle zu
Sonderpreisen
auch Vorführmodelle

Floppy Laufwerke zu **Sonderpreisen** z. B.: YD-380T + YD-180 + YD-280 + YD-380 + YD-480 + Teac-FD-55F + Floppy-Cabinet für Apple II + MF-351 Floppy 3½" von Mitsubishi

040/511 76 03
Die heiße Nr. für Ihren Computer!

IEEE-488-Karte
für IBM/PC

Kurzdaten:

- Einfache Kommandos für alle IEEE-488-Funktionen
- Schnittstelle zu BASIC, PASCAL, C, FORTRAN
- Schnell: DMA bis 800 KB/s
- Preiswert: DM 1306,— + MwSt. ab Rosenheim.

Ein ausführliches Handbuch (in Englisch) mit Tutorial über IEEE-488-Bus ist im Preis enthalten.

Lieferung ab Lager Rosenheim.

Thomatronik
Herbert M. Müller
Arnulfstraße 4a
8200 Rosenheim
Tel. 0 80 31/6 20 74
Tx. 5 25 814 muel d
BTX ✱20 298 #

Panasonic Drucker zu Tiefstpreisen!

640 KB Laufwerk für IIc **1090,-**
c't Drucker-Spooler
fertig aufgebaut incl. Netzteil und Gehäuse **340,-**
Bausatz **250,-**
TEAC FD 55 AV 1 x 40 Track .. **387,-**
TEAC FD 55 BV 2 x 40 Track .. **410,-**
TEAC FD 55 EV 1 x 80 Track .. **445,-**
TEAC FD 55 FV 2 x 80 Track .. **398,-**
TEAC FD 35 F 2 x 80 Track, 1 MB Speicherkapazität **498,-**
Erphi-Controller **298,-**
Winchester 10 MB **3998,-**

Preisliste anfordern!

UEDING electronics
Holtewiese 2
5750 Menden 1
DFÜ 02373/66877
Tel. 02373/63159

SPECIALS

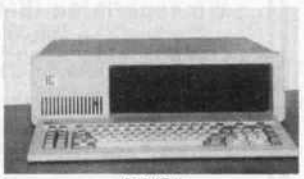
HD 64A180	70,00	Datenbücher:	
HD 64B180	85,00	TI Pocket Guide '85 Vol 1	28,50
Laborfassung f. HD	17,10	TI Pocket Guide '85 Vol 2	24,80
Q 12,288 MHz	6,00	TI TTL-Data Book Vol 1	28,50
ICL 7660	9,60	TI TTL-Data Book Vol 2	33,00
MB 1422 DRAM Contr.	35,00	Mitsubishi IC-Memories	25,00
6264-LP15	8,50	Hitachi HD64180 User Manual	18,00
41256-150 ab 25 St.	7,98		
V20-5 MHz	33,50		
V20-8 MHz	37,20		
V30-5 MHz	37,00		
V30-8 MHz	41,30		

segor electronics
kaiserin-augusta-allee 94 1000 berlin 10
tel. 030/3449794 telex 181268 segor d

Bestellen Sie bitte mit den Kontaktkarten in der Heftmitte. Versand per NN. Preise incl. MwSt. zzgl. Porto + Verpackung. Komplette Preisliste gegen Rückporto.

INDUSTRY COMPUTER Int.
ALTE LAERFELDSTR. 66
4630 BOCHUM
Tel. (02 34) 36 06 64

Herstellung und Vertrieb



IC-16/XT-H

IBM PC/XT kompatible Personal Computer mit:

- 256 KB Speicher, erweiterbar bis max. 640 KB direkt auf der Systemleiterplatte.
- 2 x serielle Schnittstelle (RS 232) als COM1 und COM2 konfiguriert.
- 2 x parallele Schnittstelle (Centronics) als LPT1 und LPT2 konfiguriert.
- 1 x Echtzeit-Uhr mit Akku gepuffert
- 1 x Game Port für Joystick-Anschluß
- hochauflösende monochrome Graphik Adapter mit 720 x 348 Punkten Auflösung, voll kompatible zu Hercules
- 2 x DS/DD Disketten-Laufwerke, Slime-Line
- 12"-Monitor, grün
- Deutsche Tastatur, IBM kompatibel
- 135 W Netzteil
- 8 IBM kompatible Erweiterungslots
- PC-DOS und Software für Uhr, Ramdisk und Printer-Spool

FÜR NUR DM 5190,—
(inkl. 14 % MWSt.)

Wir haben außerdem ein umfangreiches Programm an Zusatzkarten und Software für IBM und IBM kompatible PC.

RUFEN SIE UNS AN oder SCHREIBEN SIE UNS.
Händleranfragen sind erwünscht.

GETAKTETE NETZGERÄTE

Egermeier ELEKTRONIK
Ursula Egermeier
Viatisstraße 132
8500 Nürnberg 30
Fernruf (09 11) 40 04 08
Vertrieb Entwicklung Stromversorgung

Geringe Erwärmung, hohe Ausgangsleistung geringe Abmessungen, hoher Wirkungsgrad bis 85% es stehen 3 Temperaturbereiche zur Verfügung C 0 bis 35°C, I -5 bis +50°C, M -25 bis +70°C. Die Systeme - CM sind sekundär getaktete vergossene Module (3-Pol) - CN sind sekundär getaktete 19"-Einschübe und W primär getaktete 19"-Einschübe. Weitere Systeme sind in Vorbereitung, Sonderausführung auf Anfrage. Bitte fordern Sie unsere Datenblätter an.

Zahlung: per NN
Preis incl. Mwst.

Typ	Leistg.	Eingangssp.	Ausgangssp.	Abmessung	C	I	Preis/St.
CM 25/05 C	15 W	9 35 VDC	5V/3A	54x45x27	4%	2%	DM 115,00
CM 25/12 C	25 W	14 35 VDC	12V/2A	54x45x27	4%	2%	DM 115,00
CM 30/05 C	25 W	9 35 VDC	5V/5A	54x45x27	4%	2%	DM 136,00
CM 30/12 C	30 W	14 35 VDC	12V/2,5A	54x45x27	4%	2%	DM 136,00
CN 074 C	60 W	220 VAC	5V/5A +12V/2A -12V/0,3A	Euroeinschub 16 TE 3 HE	4%	2%	DM 370,00
	70 W						
W 061/05 C	60 W	220 VAC	5V/12A	Euroeinschub 10 TE 3 HE	4%	2%	DM 370,00
W 061/12 C	70 W	220 VAC	12V/6 A				DM 370,00

EGERMEIER ELEKTRONIK 8500 Nürnberg 30 Viatisstraße 132, Tel. 0911/400408

Die Ostereier sind da!

Jetzt billiger!
3,5" BASF 6164
1 MByte 32 x 105 x 154 mm (Blende 42 mm Option)
nur DM 349,—
Dazu die richtigen Disketten
3,5" 2 x 135 tpi
Nur Markenware wie BASF oder 3M
10 St. DM 99,—

Das flexible Laufwerk:
5,25" BASF 6138 B
Slimline, Headload, Direktantrieb, 2 x 80 Track, Spezialboard mit vielen Möglichkeiten zum Jumpern
nur DM 359,—
Dazu die richtigen Disketten
5,25" 2 x 96 tpi
Original-BASF mit Verstärkungsring
10 St. DM 59,—

Winchester
pro 1 MByte nur DM 54,30
der Beweis:
5,25" BASF 6186
18,3 MByte nur DM 999,—
Wegen großer Nachfrage Abgabe pro Kunde 1 Gerät!

Jetzt billiger!
SEAGATE 225
25,6 MByte Slimline, mittlere Zugriffszeit 85 msec
nur DM 1449,—

Maxtor 1085
85,3 MByte, mittlere Zugriffszeit 30 msec!
nur DM 5 910,90

Maxtor 1105
105,2 MByte, mittlere Zugriffszeit 30 msec!
nur DM 9 114,30

Maxtor 1140
143,5 MByte, mittlere Zugriffszeit 30 msec!
nur DM 11 143,50

Maxtor 2190
191,2 MByte, mittlere Zugriffszeit 30 msec!
nur DM 13 332,30

Superangebot:
Winchester Sub-System
18 MByte (unformatiert)
für IBM PC, Commodore PC und Kompatible
OMTI 5510 Controller, BASF 6186 Winchester, Kabelsatz und Strom-Stecker für Winchester. Kombination getestet, Winchester formatiert!
DM 1599,—

DC 512 B

(Foto vor dem Vergießen erstellt).

Subminiaturspannungswandler, paßt bei Verwendung von Spezialsockel unter 28poligen IC. Erzeugt aus 5 Volt galvanisch getrennt +12 Volt und —12 Volt bei je 40 mA, durch Fold-Back-Regelung kurzschlußfest! Wirkungsgrad besser 75 %.
DM 45,—

Versand ab DM 200,—. Preise **inklusive** MwSt. Versand an Neukunden per NN oder Vorauskasse. Lieferung solange Vorrat.

MAYON
Elektronik GmbH
Postfach 1925
8034 Germering
Tel. 0 89/84 30 51

COMPUTERSTUDIO Dieter Ulrich — Peter Runkel, Tel. 0 22 51/5 97 04 + 5 15 38; GENIE 16C, Farbkarte 256K, 2 Disk, 3999,— DM; GENIE III (gebraucht 2•80 Track) 2500,— DM; TAXAN Farbmonitor VISION EX, 660,— DM; TANDY Telefonmodem, 299,— DM; Joystick für IBM, robuste Ausführung, 59,— DM; Tastatur für IBM Cherry, 190,— DM; IBM-Zeichensatz für EPSON Drucker, 139,— DM; V20-Chip (70108) für 8088 Super!!!, 135,— DM; günstige Laufwerke und Epson Produkte. G

SHARP PC1401/2 SPEICHERERWEITERUNGEN: auf **10KB** (99 DM), auf **18KB** (189 DM), mit Garantie! PC1260 auf 10KB (99 DM), andere Typen auf Anfr., NEUGERÄTE, z. B. PC1401 mit 10KB: 289 DM! Einbau ca. 1 Woche, V-Scheck oder Nachnahme. **Schillings,** Entenbachstr. 16, 8000 München 90. G

c't-Klangcomputer, 7 leere Voice-RAM-Karten f. **DM 210,—** zu verkaufen. Tel. 0 40/27 81 57.

IBM-kompatible PCs, incl. 256K, 1 Floppy (Markenfabrikat), DIN-Tastatur, 155-Watt-Netzteil für **1690,— DM.** Tel.: 05 21/2 08 00 07, ab 17 Uhr.

ATARI ST, C-Compiler von **GST** (Test c't 2/86) mit GEM-Editor, Assembler: 300,— DM (Englisch). Konrad Mayer, Zuger Str. 4, 8000 München 71.

Fachzeitschr. per Jahrgang zu verkaufen: Electronics 74—78, Elektronik 76—84 DM 2,—/St., El. Design 79—85, EDN 79—84, Dig. Design 74—76, Comp. Design 77—85, MM Syst. 83—85 DM 1,—/St. Tel. 0 70 42/7 82 11, nach 17 Uhr.

NEU + **Jetzt auch in Köln!** KELM + HOMBERG electronic, Hohenstaufenrg. 43—45. **Neueröffnung am 12. 3. 86. Super Sonderangebote!** G

Datenkabel, Standard oder nach Ihren Angaben; Steckverbindungen, Zubehör. **Kleinmengen- und Eilservice!** Info und Preislisten kostenlos. WERNER WINKLER ELECTRONIC, Zum Loh 16/1, 2725 Kirchwalsede. G

Künstliche Intelligenz Literaturverzeichnis 12,— DM von Knowledge Ware. Postscheck-Konto 286080-600 FfM.(steuerl. abzugsfähig). G

Verk.: **BASIS108/128K,** 2 Org. APPLE Laufwerke, Mon. 20 MHz, HGR1024 Grafik-Karte mit Softw., 256K-Pseudo Disk ohne Softw., APPLE 3 Monitor, Drucker Epson MX80 umgerüstet auf HGR, Softw. DOS3.3, UCSD-Pascal, CPM2.2, Turbo-Pascal, Textverarb., Datenbank-System, Handbücher. Zusammen VHB DM 5000,—, unter C860402.

c't 86 CPU, I/O, Floppy-Contr., 256-KB, Bus, **getestet** + CP/M 86, **VB 1080,—.** Tel. 05 21/45 09 32.

IBM-Farbgrafik-Karte für Commodore PC10-Monitor DM 450,— incl. Umbauanleitung. Tel. 0 51 31/9 55 00-01.

Suche CP/M Comp. Fligge, Kunoldstr. 22, Kassel.

Bestücken, Löten und Montieren von Platinen, Baugruppen, Mustergeräten und Kleinserien. Sonderanfertigungen und Reparaturservice. Schnell und zuverlässig. DETRONIK, Westring 117, 4830 Gütersloh, Tel.: 0 52 41/5 55 59. G

SCHNEIDER CPC. Interessante Programme für die Schneider-CPC-Computer erhalten Sie bei Friedr. Neuper, 8473 Pfreimd, Postfach 72. Fordern Sie einfach das kostenlose Info an. Kommt sofort! G

WAHNSINN!!! Osborne Executive (komplett bestückte Platine) mit org. Tastatur u. **Riesensoftwarepaket:** CP/M 3.0, Wordstar, Supercalc, Pascal, C/M Basic, M80/T80, L80, Mailm., ZSid, Helps, Utilities, Bücher usw., VB 1500 DM. ANDREAS THUN, 8000 München 60, Mooswiesenstr. 3a, Tel.: 0 89/83 28 69.

BASF 6106, SS, 40 Track, neuwertig, zu verkaufen, **200 DM.** Tel. 0 24 21/5 88 71, nach 17 Uhr.

c't 68K EBC, 512 KByte DRAM, Graphic, Busplatine, aufgebaut und getestet, für 2500 DM. Tel.: 05 11/45 15 41, abends.

ZX81-Assembler gesucht. 0 30/4 15 28 39 ab 18 Uhr.

APPLE //c, 128K, 2 Disks, neu, originalverpackt, VHB 2800,— DM (neu 3400,—). 02 21/89 73 00 ab 18 h.

CHERRY G80-320 parl. neu + ungebraucht zu verk., VB 550,— DM. 0 94 44/81 37, 18—19 Uhr: Mo.—Fr.

Heath/Zenith WH-89 H-37 Softs. Controller, **H-14 Drucker,** Pascal MT+, MS-Basic Compiler, M80/L80 Macro Assembler, SID/ZSID Debugger, Handbücher, Z-80 Literatur, dt. Zeichensatz, DM 2250,—, unter C860401.

★★ WaveMate Bullet — das CP/M-Geschoß ★★ CP/M+, 1x8" (1,6 MB), 1x5" (1 MB), Terminal Ampex, alle Software, Zubehör, VB 3900 DM. U. Schmiedeskamp, Tel. 02 41/2 72 66.

Profifotokopierer OCE-Elektrostatik 1400 mit Verkleinerung DM 350,—. **70 cm Handfunke,** Yaesu FT 708 R, Handbuch, Zubehör, DM 550,—. **Apple III + Profile,** Software, (II er Emulation), VB DM 3700,—. **Sharp MZ 80 B** Gr I + II, Cen. Interf., Floppy, viel Software, VB DM 3250,—. **Casio FP 200** VB DM 370,—. Tel. 0 26 41/14 41.

NEU! PLANTRON PC16-AT NEU! PLANTRON PC16-AT NEU 80286 6MHz, 640kRAM bis 3,2MB erw. 4ROM Sockel BIOS instal. Parallele u. serielle Schnittstelle 200W Netzt., dt. Tastatur, 8I/O Erw.-Slots, Grafikkarte, Betriebssystem MS-DOS, Floppy-Festplatten Control. + 1,2MB Floppy SUPER **6998,—;** +25,6MB Festplatte UNFASSBAR **8998,—;** und das alles noch + Monitor 12", 6 Monate Vollgarantie u. v. Vertragshändler. Zugreifen. Wir bieten auch Finanzierung u. Plantron XT's. BHK-Elektronik, Anita Kattner, Klausenburgerstr. 166, 6100 Darmstadt, 0 61 51/31 52 98 ab 17 Uhr a. SA/SO. G

C64 ★ Handwerk ★ Bauwesen ★ Architektur ★ C64 Leistungsstarke Programme für den 64: Angebot/Rechnung/Bestell-/Lieferschein ★ Rechnungsausgang/Zahlungseingang/MwSt. ★ Mahnung ★ Preisspiegel ★ Lv für viele Gewerke auf Diskette lieferbar. **!TS ★ K. HERWEG, Postf. 60 02 20, 4630 Bochum 6!** G

MiTeA GmbH — Adrianstr. 6 — 5800 Hagen 1 Tel. (0 23 31) 7 00 17/8 ★ EDV-Zubehör und dig. Bauteile aus Produktionsbeständen Europrint FT80X DM 695 ★ Tastaturen Floppy/Winch-Laufw. ★ FUJI-Disketten 4164-150 ab 10 Stck. 3,60 ★ ECB-Karten ★ Lagerliste mit Rückumschlag anfordern! G

CTK Reise-Kommunikations-Set 4

im eleganten, robusten Attaché Doppelkoffer.

Für Außendienstmitarbeiter, Journalisten...
Mit **CTK Profi-Akustik-Koppler** und **NEC** handheld-**Computer 8401 BD**
Das tragbare Büro
Außendienst-Komplettlösungen inclusive Textverarbeitung, Kalkulation, Dateiverwaltung, Kommunikation.
netzunabhängig, reisefertig.

CTK
CTK Computer-, Text- und Kommunikations-Systeme GmbH
Dolmanstraße 82 5060 Bergisch Gladbach 1
Tel. 0 22 04 / 6 30 61 · Telex 8 873 742

KLICMAN – KÖLN
FACHBETRIEB DER ELEKTROINNUNG KÖLN

Wir reparieren APPLE und kompatible Computer
Karten, Floppys, Tastaturen etc. — preiswert, zuverlässig
Wir reparieren IBM und kompatible Computer
(nach vorheriger tel. Anfrage)
Entwicklung, Beratung, Service, Verkauf.
Olivetti M 24, SPERRY,
Tulip System PC's
Apple-komp. und IBM-komp.
Fordern Sie noch heute unsere Preisliste an!

KLICMAN Elektronik Mikrocomputertechnik
Moltke-Straße 32 · 5000 Köln 1 · Tel. (02 21) 24 12 23

IBM-kompatibel
NEUE PREISE

XT Mainboard 256 359,—
Hauptplatine mit 8088 CPU. 8 Slots mit vollen 256 K-RAM

ICO 360 1797,—
Rechner m. XT-Mainboard 256, Colorkarte, Monitoranschluß, 1 Disk Drive a 360 KB, Tastatur.

ICO 720 2099,—
wie ICO 360, jed. m. 2 Disk Drives m. zus. 720 KB

ab 2290,—

6 Monate Garantie.
Versand erfolgt per NN oder Vorkasse.

Klaus Jeschke
Hard-, Software
Viertstr. 3—16
6233 Kelkheim
☎ (06198) 7523

WENZEL ELECTRONIC
OKIDATA OKIDATA

Teile Nr.	zugr. nS	1 stück	ab 10
RAMS			
HM6116LP3	150	6.80	6.45
HM6264LP15	150	9.45	9.00
UPD4364L-15	150	9.45	9.00
MK2064-5	150	9.45	9.00
UP041256	150	8.95	8.50
EPROMS			
D2732A	250	8.20	7.80
D2764A-3	300	4.90	4.65
D2764A-2	200	6.80	6.45
D26C64A-2 C-MOS	200	12.75	12.10
D27128	250	8.50	8.05
D27256-2	200	15.40	14.65
D27C256-2 C-MOS	200	25.00	23.75
uP Bausteine			
Z80A CPU C-MOS	Z84C00	19.35	18.40
Z80A CTC C-MOS	Z84C20	15.35	14.60
Z80A PIO C-MOS	Z84C30	15.35	14.60
80C39 C-MOS	CPU	9.50	9.05
80C31 C-MOS	CPU	29.00	27.55
UPD70108 V-20	CPU	39.95	37.95
UPD70116 V-30	CPU	44.95	42.70
P8052AH-BASIC		69.00	64.55

OKIDATA Drucker		
Microline 182 parallel		1049.00
Microline 192 parallel		1699.00
Microline 192 parallel mit Einzelblatteinzug		1999.00
Traktor und Schallschluckhaube	zus.	79.00
V24/RS232C Interface	aufpreis	228.00

Liefer und Zahlungsbedingungen:
Alle Preise inklusive 14% MwSt. Porto/Verpackung 5.80 DM
ab 200 DM Auftragswert frei. Lieferung per Nachnahme.
Mindestauftragswert 30 DM. Alle Angebote freibleibend.
Zwischenverkauf vorbehalten.

Wenzel Electronic 8751 Niedernberg
Hauptstr. 95 Tel. (06028) 7883

Parallel-Matrix Drucker ITOH-8300P/SP wie neu, NP 1700,— f. DM 700,—, Elzet 80 Tastatur, Gehäuse, Kabel f. serielle Schn.St. neu DM 200,—, div. Bauteile, IC's, Bücher. Tel. 0 50 73/6 06.

PROGRAMMIERER — Durch Fernkurs zu fundierten Kenntnissen als BASIC- oder COBOL-Programmierer. Ohne besondere Vorbildung werden Sie Computer bedienen, Programmsprachen beherrschen, Programme entwickeln, testen und anpassen können. Als zukunftsorientierte berufliche Weiterbildung oder interessante Freizeitbeschäftigung. 48 weitere Fernkurse. Kostenlosen Studienführer und Probelektion anfordern! Kein Vertreterbesuch. **Studiengemeinschaft Darmstadt,** Abt. 31/99, Postfach 41 41, 6100 Darmstadt. G

++++ **Probleme mit Netzstörungen** ++++ Bringt Ihre Computeranlage unerklärliche Fehler oder stürzt gar das Programm ab, liegt dies in vielen Fällen an **verseuchter Netzspannung** (spikes). Als Basisschutz gegen Überspannungen sollten daher alle Computer mit **Steckdosenfiltern sf 220** ausgerüstet werden. Lieferbar in 4A od. 6A ab DM 56,95. **Manger KG, Postf. 1127, 7800 Freiburg, 07 61/5 05 38.** G

COMPUTER-ZUBEHÖR — kostenlose Info anf. z. B. 10x Disk 1D 28,—, 10x Disk 2D 39,—. K & N, Postfach 90 08 06, 2100 Hamburg 90, Tel. ☎ 040/7 63 13 65. G

PC 1500/A, int. Vollausbau des Basic-Speicher bis zu **28KB bei frei belegb. Modulfach.** Einbau einer durch Softw. gest. Portschltg. zur Verdopplg. d. Lade-, Druck-, Rechengeschw.; Super Softw. u. **Selbstbaumögl.;** kostenl. Info anf.: **H.R.M.-electronic, Rückertstr. 23, 4650 Gelsenkirchen, Telefon: 02 09/87 77 98.** G

PC 1401, interner Vollausbau auf 10 KB DM 90; neu eingetr.: Selbstbau mit **8KB Flat-Pack-RAM HM6264LFP15** DM 48. Kostenl. Info anfordern **H.R.M.-electronic, Rückertstr. 23, 4650 Gelsenkirchen, Telefon: 02 09/87 77 98.** G

NEU ++++++++++++++++++++ **Jetzt auch in Köln!** KELM + HOMBERG electronic, Hohenstaufenrg. 43—45. **Neueröffnung am 12. 3. 86. Super Sonderangebote!** G

Commodore PC10: Adreßdecoder-PAL DM 80,—. Tel. 0 51 31/9 55 00-01.

★★★ **SCHRITTMOTORINTERFACEKARTE** ★★★ ★ **XYZ-Achsensteuerung** für alle Computer mit ★ Parallelschnittstelle. Kompl. mit Netzteil und 3 Schrittmotoren ***DM 259,—, SCHRITTMOTOR einzeln ***DM 24,—, BOHRPROGRAMM C** 64/Disk DM 98,—, Info DM 2,—. **PME,** Hommerich 20b, 5216 Rheidt. Wir übernehmen **CAD-Layout Entflechtungen** auf IBM/HP sowie **Bestückungen** (autom. Lötanlage). G

An dieser Stelle könnte Ihre private oder gewerbliche Kleinanzeige stehen. Exakt im gleichen Format: 8 Zeilen à 45 Anschläge einschl. Satzzeichen und Wortzwischenräumen. Als privater Anbieter müßten Sie dann zwar 31,92 DM, als Gewerbetreibender 52,90 DM, Anzeigenkosten begleichen, doch dafür würde Ihr Angebot auch garantiert beachtet. Wie Sie sehen.

NIEDRIGSTPREISE für EPROMs und RAMs; z. B. 27128-250 nur **11,—** ★ 2764-250 nur **9,95** ★ 27256-250 nur **39,50** ★ 2114-250 nur **5,50** ★ 2114-200 nur **6,25** ★ 4116-200 nur **3,95** ★ 4116-150 nur **5,10** ★ 4164-200 nur **4,95** ★ 41256-P20 nur **17,95** ★ 41256-P15 nur **18,95** ★ 6116-LP3 nur **6,95** ★ **Weitere Bauteile auf Anfrage!** Versand per NN zuzügl. Versandkosten oder gegen V-Scheck 3,— Versandk. ★ Fordern Sie unsere neue kostenlose Sonderliste an! **R. Rohlederer, Saarbrückener Str. 43, 8500 Nürnberg 50, Tel. 09 11/48 55 61.** G

IBS-RAM DISK AP 17 MIT 256 K, SOFTWARE: DM 380; **IBS Z80 H, 9 MHZ,** SYSTEM + DISK PATCH: DM 400; EHRING FDC 4 MIT KABELSATZ: DM 130; G. NEUMAIER, EICHENSTR. 2, 7550 RASTATT, TEL.: 0 72 22/6 03 56.

CEPAC-65, neuwertig, incl. EPROM 2732 (Interface CE50/60), DM 100,— VB. N. Mohr, Einsteinstr. 6/6, 7460 Balingen 12.

CP/M 2.2 + 3.0 Computer, PROF/GRIP, 128KB, 2LW 80Tr DD/DS, MAKII Tastatur, Monitor 22MHz/gr. Text-Daten-Kalkulation, Pascal, Fortran, Basic, Cobol, C, Assemb., Eprommer, VB 3000,—. 0 30/7 85 18 37/4 91 32 04.

Verkaufe günstig NDR-KLEIN-COMPUTER-PAKET1 (Z80). Tel. 0 89/3 51 04 06 nach 18 Uhr.

AP-102 mit 2 MHz 68B09CPU inkl. 64KDRAM + FDC + PIA + Graphic Interf. 512x256 + Realtimeclock + ACIA + Floppy DS/DD/40Track + Tastatur. Software: Flex9 Assembler, XBASIC, Dynacalc (wie Visicall), Stylograph und v. mehr. Teilweise CT-82 Emulation. Alles für nur 1600,— DM VB. Tel. 02 14/9 51 68.

HILFE! Wer besitzt Hefte (möglichst viele) der Zeitschrift 'Prisma' — das ist die Clubzeitschrift des CCD-HP41 User Group — und kann mir diese kurzzeitig zur Verfügung stellen! Versandkosten werden Euch erstattet! Bitte erst Kontakt mit mir aufnehmen! A. Müller, Reinhold-Frank-Str. 25, 7500 Karlsruhe.

Apple 2 +, comp. + 80Z + 128K + Mon. + Disk + Contr. + Prommer + Druckerinterf. + abges. Tast. + Stahlblechgeh. + Software ****VB 2000,— DM**** Tel. 0 71 56/2 75 89.

Verkaufe CPM Rechner, v. Extras. 0 51 74/17 95.

★★★★★ Apple IIe/+ ★★★★★ 512K RAM-DISK für DOS + CP/M + UCSD. !!!! Rasend schnell !!!!! Leerplatine + Software 99,— DM; bestückt (ohne Ram's) 199,— DM. Norbert Menke, Tel.: 0 52 51/2 58 87, Kilianstraße 32, 4790 Paderborn.

Verk. 8"-Floppy-LW. BASF 6101 m. Netzteil im Gehäuse, geeignet zum Booten z. B. c't 180, für DM 150,—. Tel. 0 45 34/74 41 (ab 18 Uhr).

GRIP-2 getestet, VB 390, div. **ECB-Systemkarten** (RAM, CPU, SIO, etc.), Preis VS, **Televideo TVI 950** Terminal, VB 1200, Netzteil, Gehäuse. M. Haase, Tel. 0 60 62/75 45.

ITT 3030 mit viel Software zu verkaufen. CP/M 64 k RAM, 2•560 k Laufwerke u. Monitor, dt. WordStar-Tastatur, ideales Bürogerät incl. WordStar, DBasell, Mailmerge, Spellstar und Basic, DM 2500,—. U. Heller, Im Tal 29, 7742 St. Georgen 3, Tel. 0 77 24/25 53.

MACRO CROSS ASSEMBLER für CP/M und MSDOS Rechner, 6800, 6801, 6805, 6809, 68HCII, 65C02, 6303. Donald W. Corson, Am Kopf 24, 5900 Siegen, Tel. 02 71/8 88 15. G

Computer-Preisliste

IBM©compatibel

Alle Preise gelten solange Vorrat reicht.

Bezeichnung	Beschreibung	Bestell-Nummer	
Color-Card	16 Farben, Chinch-Anschluß FBAS-BAS, RGB Buchse, 640 • 200	25-018	175,–
Netzteil 15 Amp.	Schalter seitlich, geräuscharm	25-012	219,–
Monitor	12" Bernstein, Video Chinch, entspiegelt 22 MHz	25-140	289,–
640 K-Board	8088 Mainboard aufrüstbar bis 640 KB, 8 Slots, XT-kompatibel 0 KB	25-027	388,–
Laufwerk	2•40 Track, 0.5 MB, 360 KB formatiert	25-038	350,–
Drucker	180 Z/s.(NLQ 18×20), Walze + Tracktor, Centronics	25-065	1.189,–

HÖSCH ELEKTRONIC
Bruchstr. 43
4000 Düsseldorf 1

Gratis-Unterlagen anfordern

Preise incl. MwSt.

Mikrocomputer-Reparaturen

(Preise inkl. Bauteile)

IBM PC
Motherboard ..... DM 150,—
Interfacekarten ..... DM 100,—

Apple II
Motherboard ..... DM 150,—
Interfacekarten ..... DM 50,—
Diskettenlaufwerke ..... DM 120,—

C64
Rechner ..... DM 150,—
Laufwerke ..... DM 120,—

Druckerreparaturen (z. B. Epson, Star) sowie Reparaturen anderer Geräte auf Anfrage.

P.S.: ROLL-MAUS für PC, Apple II, C64 ..... DM 195,—

Dipl.-Math. W. Dederichs
Büro für Software- und Hardware-Entwicklung
Hackstückstr. 11, 4320 Hattingen, Tel.: 0 23 24/5 22 40

INTUS -Lern- und Anwenderprogramme

für Apple II und IBM-PC und Kompatible

- Business-English und Alltags-Englisch
- Maschineschreiben wie der Blitz
- Basic- Lernprogramm, ausgezeichnet
- Rechtschreibtrainer deutsch und englisch
- PriBu — Privatbuchhaltung
- Unternehmens-Planspiel 'Markt + Gewinn'
- Wissenstrainer (eig. Fragen u. Antworten)
- und über 200 weitere Programme. **Katalog** gratis
- Demo-Disketten mit 5—9 Teilprogrammen **DM 10,—**
- **14 000** Frei-Programme (fast) gratis Programm-Liste (Vorkasse) **DM 10,—**
- Bei Bestellungen bitte System angeben.

INTUS SOFTWARE
Kaiserstr. 21, 7890 Waldshut-Tiengen, Tel. 0 77 51/79 20

RAIL-electronic GmbH

Auszug aus unserem Liefer- u. Lagerprogramm
1. Wahl, Industriequalität

2764-250 ns	**7,90 DM**	74LS00	**—,60 DM**
27128-250 ns	**10,50 DM**	74LS14	**—,80 DM**
27256-250 ns	**19,90 DM**	74LS32	**—,60 DM**
4164-150 ns	**3,65 DM**	74LS125	**1,20 DM**
41256-150 ns	**9,90 DM**	74LS153	**1,20 DM**
6116 LP-3	**7,90 DM**	74LS161	**1,30 DM**
6264 LP-15	**13,50 DM**	74LS166	**2,90 DM**
8087 C-2 8 MHz	**530,00 DM**	74LS640	**2,60 DM**

HM 3-65161-5 RAM stat. 2K x 8, 70 ns
f. ct-Bauanleitg. i. ct1 + 4/85 ab Lager lieferb. Prs. a. Anfrage!

Wir führen lagermäßig 74 LS, S, HC, F . . . Serie
Epson + Star-Drucker Preis auf Anfrage
DIGITALE Bausteine **sind unsere STÄRKE!!**

RAIL-electronic GmbH, Großer Biergrund 4,
6050 Offenbach/Main, Tel.: 0 69/88 20 72, Tx. 4 152 890

AUS EIGENER ENTWICKLUNG: SER. SCHNITTSTELLE AS1000 FÜR CPC 464/664-6128 (2·V24, O. EXT. STROMVERSORGUNG) AB 189 DM ** SONDERPREISE FÜR LEERPLATINE ODER BAUSATZ!! COMPAC-TERMINALPROGRAMM ZUR AS1000 NUR 59 DM ** GRIPS-UMBAUSATZ FÜR GRIP-MEHRFACH SCHNELLER — KEINE BILDSTÖRUNGEN MEHR — NUR 99 DM ** SOFTWARE ZU GRIPS — GRIPSCOM TERMINALPROGRAMM NUR 99 DM ** SONDERPREISE BEI KOMPLETT-PAKETEN: Z. B. GRIPS + GRIPSCOM NUR 169 DM ** KOMPLETTE DFÜ-PAKETE F. CPC UND GRIPS ** UMKOPIEREN VON DISKETTEN, PREISE UND FORMATE AUF ANFRAGE ** ALLE PREISE INCL. MWST., ZUZÜGL. VERSANDKOSTEN ** A. KRISCHER, ELEKTRONIK-VERSAND, NOPPIUSSTR. 19, 5100 AACHEN, TEL. 02 41/3 28 96. G

APPLE + IBM kompatible SYSTEME + KOMPONENTEN ERPHI Autopatch Controller 288,—; PHILIPS X3134A 429,—; TEAC 55F 449,—; DATAPHON s21d 289,—; TAXAN Monitore, Disketten SKC DS/DD/QD 10 Stck. 67,—; IBM komp. Systeme aus hochwertigen Komponenten, z. B. Festplattensystem 10 MB, kompl. einsatzfähig mit gr. Tastatur und 25 MHz Monitor ab 4998,—; Motherboard XT 256K, Multifunktionskarten, Festplatten 10—80 MB, Harddisk-Controller, Gehäuse, Netzteile, Tastaturen, . . . kostenl. Gesamtliste mit vielen anderen Artikeln von DIETMAR TEICH DATENTECHNIK, Queller Str. 94, 4800 Bielefeld 14, Telefon: (05 21) 45 09 32. G

Sinclair ZX81 Bausatz mit 16K-RAM-Modul + deutschem Handbuch + Bauanleitung, komplett DM 98,—; einzeln ZX81 DM 65,—; 16-k-RAM Pack DM 49,—. Computer Accessoires Int'l GmbH, Tel. 0 89/ 6 09 36 07. G

GRIP-2 Graphik Systemkarte ECB-Buss. Fertig aufgebaut und getestet, für nur DM 400,—. M. Zimmermann, Tel. 0 52 55/18 73 nach 17 Uhr.

HAMEG Oszilloskope — Tastköpfe — sofort ab Lager — Koaxialkabel maßkonfektioniert — Ruf 0 42 98/ 49 80 — Kamera f. Oszilloskope u. Monitor — Göbelstr. 54 — BACHMEIER-electronic — 2804 Lilienthal. G

TREIBERPROGRAMM für PROMMER-80 (c't 2/85), in **Z80-Assembler,** programmiert **alle 27xxx bis 27256, erweiterbar,** 3 Programmieralgorithmen, ausführl. Handbuch, DM 60,— + Porto. K. Becker, Lerchenkamp 23, 2000 Hamburg 61, 0 40/5 50 55 78. G

Für **6809-Systeme,** speziell **EUROCOM-2: XBASIC,** Grafik **GRANEW** (für BASIC + PASCAL), **TOOLKIT** zu verkaufen (Original!). Tel. (0 71 51) 6 33 84. G

SS-100 Busplatine mit 6 Steckplätzen, 4 Stück vorrätig, Preis VB. 0 21 62/3 08 26, öfter anrufen.

Z80 fig-FORTH (CP/M) frei geg. form. 8"- od. 5¼"-Disk & Rückporto. E. Ramm, Pf. 38, 2358 Kaltenkirchen, (0 41 91) 16 21.

TEKTRONIX – 465 SCOPE 200 MHz Dualtrace 2x10:1 Probes zu verk., VB 2000,—. Tel. 0 70 42/ 7 82 11, 17 Uhr.

GRIP-2.5 + PROF 80 4MHz 128K RAM + **Monitor** 18MHz + ECB-Bus 64pol. + Tastatur, zusammen **1500 DM.** Ralf Losse, Dorfstr. 18, 2059 Müssen, Tel. 0 41 55/34 57 ab 18 Uhr.

Ärger mit Vizawrite? Umlaute für MPS 801, VC 1525, DPS 1120. Infizes für Star-Drucker? Ähnliche Probleme? W. Fahl, Friedrichstr. 32, 5100 Aachen. G

Triumph-Adler Alphatronic PC mit Doppelfloppy 3½ Zoll (2x640 K), Sanyo Monitor bernst., Software (z. B. TA-Dateiverw., Brief- u. Adreßprogr., Logicalc), 10 Handbücher, 30 Disketten 3½ Zoll 2D (Sony) um DM 2500 zu verkaufen. Tel. 07 11/76 14 42 (nach 19 Uhr).

c't 68000 Komplettsystem (SBC), getestet incl. Netzteil, Gehäuse, 1 Laufwerk, Tastatur, Monitor (or) und Literatur, zusammen nur 4000,— DM. Tel.: 0 42 21/ 8 87 76 (nach 18 Uhr).

GRIP-1 aus c't 6/84, VB 500,— DM. Tel.: 8—15 Uhr 02 31/1 08 41, ab 19 Uhr 0 25 92/59 99.

Suche **8" Winchester** LW SMD Schnittst. und CPM Einplatinen-Rechner mit RAM und FD-Controller für 8" FD. **Suche** für **DEC** PDP 11 Module. . und. . und. . alles! Tel.: 0 89/71 87 11.

Suche 8"-Station für PROF 80 (Shugart-Bus, kein Controller). H. Theer, Eckermannstr. 14, 3500 Kassel.

NEU + **Jetzt auch in Köln!** KELM + HOMBERG electronic, Hohenstaufenrg. 43—45. **Neueröffnung am 12. 3. 86. Super Sonderangebote!** G

40Z/s schneller Typenraddrucker Diablo-Farbbänder u. Typenrad, programmierbar, 36 ESC-Codes, Centronics-kompatible Schnittstelle (8 Bit), solange Vorrat reicht, nur 1498,—. Tel. 0 30/7 05 67 65. G

JANICH & KLASS LEERPLATINEN CPU II; FDC II BUS 9 **EPSON** 3,5" LW, **HALTEC** NETZTEIL; alles fabrikneu u. unbenutzt, inkl. Dokumentation; kompl. abzugeben. Tel.: 02 41/52 86 05 ab 20 h.

COBOLD Entwicklungs- u. Steuerungsrechner, siehe elrad 2,3,4/84, erweiterte Version, inkl. NT u. Handbuch. Tel.: 02 41/52 86 05 ab 20 h.

SCHNEIDER-CPC- UND SPECTRUM-HARDWARE: Sound-Sampler, EPROM-Brenner, Schrittmotor-, Floppy-, Drucker-, RS-232-Interfaces, Echt-Zeituhr, D100, RAM- und EPROM-Erweiterungen direkt vom Hersteller: Gratis-Info anfordern! MICRO-DATENSYSTEME, 8723 GEROLZHOFEN, PF. 13 25. G

Fat Mac Speichererweiterung für **Macintosh** von 128KB auf 512KB 700,— DM. Schuh Pia, Promberg 1, 8122 Penzberg, 0 88 56/77 09. G

APPLE Comp. 16k, 80Z, Z80-Karten Centronics Interface Sanyo Monitor 2 Apple Drives progr. Eltec Tastatur (Datastar-Wordstar angepaßt), Software Handbücher, Preis VB 1950. 0 24 63/87 27.

Spectrum 48k + Interface I + Microdrive + Car Software (Tasword, Beta Basic, Devpac etc.), Lit. (ROM-Listing etc.), Preis VHS. 0 24 63/87 27.

Suche **ACORN B** mit Diskettenstation, Zweitprozessor 6502, Monitor, Drucker, Forth-Compiler. Jennet, Bienwaldring 4, 1000 Berlin 47.

SPECTRUM 48k Software: 2-Passass., MDV-komp., >100 Zeilen/s, komf. Editor (24 Bef.) + Disass.-Debug. (u. a. TRACE), beide Prg. verarb. alle inoff. Bef., zus. DM 50 inkl. ausf. Anl. **Schnelles SAVE & LOAD** (6000 Baud) mit einf. Basicbef. nur DM 20 ★ Info DM 1 (Marken) bei: M. Stramm, Rütscherstr. 155/ 1513, 5100 Aachen. G

die **TRAGBAREN**
Stückpreis unter DM **2.500,—**
Lieferumfang: **CP/M** und von **MicroPro** WordStar/MailMerge, CalcStar, Info/Star

SIG COMPUTER GMBH
6333 Braunfels 4
Tel. 06445/804
Telex 483800 sigma

PROTEUS 68/20 . . .
Der Maximikro mit der MINI-Leistung!
Die Leistung der MINI-Klasse zum Preis eines Mikrorechners.

- **MC68020** Prozessor
- **MC68881** Gleitkomma-Coprozessor für superschnelle Arithmetik
- echte **32 Bit** Struktur, bis **4 MByte RAM** im Grundsystem
- **VME-Bus**
- **für 16,67 MHz** Taktfrequenz ausgelegt (bereits lieferbar in der 12,5 MHz Version)
- 1 MByte (brutto) Floppy, bis 150 MByte (netto) Festplatte, bis 40 MByte Streamer (opt.) integriert
- bis 20 Arbeitsplätze anschließbar
- on-board Grafik

Extreme Rechengeschwindigkeit und Flexibilität – dafür sorgen Coprozessor, **VME**-Zusatzplatinen und ein breites Spektrum verschiedener Betriebssysteme vom Einplatz- bis zum **Multiuser-Multitasking-Realtime**-System.

OS-9®, Concurrent DOS 68K®, EUMEL/ELAN, CP/M 68K® und UCSD-p-System®

Der Cache des MC68020 wird von allen Betriebssystemen genutzt, der Coprozessor durch höhere Programmiersprachen wie **FORTRAN, C, ...** unterstützt.

Die Aufwärtskompatibilität vom Supermikro PROTEUS 68/10 zum Maximikro PROTEUS 68/20 erlaubt die Übernahme von Programmen und Dateien. Ein markanter Vorteil durch die Kontinuität der **PROTEUS**-Linie!

CeBIT Hannover Messe '86
CeBIT — 12.—19.3.86
Halle 17 • Stand 204

PROTEUS GmbH
D-7500 Karlsruhe 1
Haid-&-Neu-Straße 7
☎ (0721) 69 30 15
Tx 7 826 349 prot d

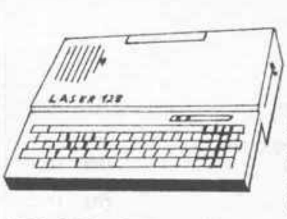
Spectrum-Eprommer
248,—

- ★ 2716/ 2732/2732A/ 2764/27128
- ★ Intelligente Programmierung
- ★ Menue — gesteuert
- ★ zus. Centronics Schnittstelle
- ★ Keine zus. Hardware erf.
- ★ Preis **248,— DM** incl. Software für EPROMMER und Drucker
- ★ Spez. Sockel zum Austausch Spectrum-ROM → 27128 **24,80 DM**
- ★ EPROM-Löschgerät **117,00 DM**
- ★ Druckerkabel Centronics **37,00 DM**

Alle Preise incl. MwSt. zuzügl. Versand, Info anfordern.

C&M Meyer, Rahserstr. 52, 4060 Viersen 1, Tel.: 0 21 62/2 29 64

pyramid computer
tel. 0761/66843 telex 772522 pyram
am Galgenberg 15 7800 Freiburg



LASER 128
128 kb ram
apple //c kompatibel
143KB Laufwerk,128KB Ram,80 Zeich
Video 16 Farben,560 x 384 pixel,
2x serial,parallel,Mouse Schnitt.
Z 80 CP/M optional. **Nur DM 1850.-**

IBM & APPLE kompatible produkte zu! interessanten preisen. ⟶ fordern sie unser hardware kompendium an«‹

apple und ibm sind eingetragene warenzeichen

NUTZEN SIE DBASE II/III NOCH PROFESSIONELLER! ★ Die Programmierertools **DBASEF** bieten Ihnen ★ **DBASEF** komprimiert Programme → **PLATZ + ZEIT** ★ **DBASEF** formatiert Programme → **ÜBERSICHT** ★ **DBASEF** überprüft die Syntax → **SICHERHEIT** ★ Menu- und Kommandoaufruf / dt. online Hilfe / Parametrisierung . . . für CPM(Z80), IBM-Komp. NUR 98,—. **Info und Lief.: SOFTWARE + BERATUNG** B. Schillingstift 42, 2000 Hamburg 55. G

3 000 000,— GEWINNEN??? Softcontrol Lotto-Prg. <LOT2000>. Alle bisherigen Ziehungen abgespeichert. Auswertungen und Ziehungen automatisch. Wird auf Apple-2 CP/M-Diskette geliefert. Preis komplett mit P/V DM 29,90. Manfred Filbrich, Hauptstr. 22, 3171 Adenbüttel, Tel. 0 53 04/17 64. G

DISKKONVERTIERUNG über 120 Formate, Endpreis 15 DM. INFO Tel. 0 23 51/7 82 21. G

!!!!!!!!!!!!**FORTH** läuft auch auf Ihrem Micro!!!!!!!!!!!! Einfach das Assemblerlisting eingeben, und Sie haben Ihren eigenen FORTH-Compiler. Kostenl. Info bei: B. Lipgens — Datentechn., St.-Florian-Str. 20, 8399 Ruhstorf. G

HITRANS AKUSTIKKOPPLER m. FTZ-Nr.; 300C DM 298,—; 1200 Bd., BTX-Vers. od. 300/1200 Bd. umschaltb. DM 598,—. **MIO DEVICES,** Laibnerstr. 3, 8919 Utting, 0 88 06/17 52. G

PC voll IBM-kompatibel ab 2000 DM incl. MwSt. INFO Tel. 0 23 51/7 82 21 tägl. ab 18 h u. SA/SO. G

Commodore PC10: Adreßdecoder-PAL DM 80,—. Tel. 0 51 31/9 55 00-01.

IBM-Farbgrafik-Karte am Commodore PC10-Monitor DM 450,— incl. Umbauanleitung. Tel. 0 51 31/9 55 00-01.

WAHNSINN ← LOTUS 123, 1331,—; WORDSTAR D., 807,—; FRAMEWORK, 1398,—; SIDEKICK II, 215,—; SYMPHONY D., 1827,—; WORDST. + MAILM, 1032,—; SUP. PROJECT, 1227,—; SOFT. KATALOG, 5,—; SUCHE LFD. PROGRAMME, AUCH VON PRIVAT; ALLE PREISE ZZGL. MWST. SOFTWARE WRONNA, BERGHEIMER STR. 28, 5010 BERGHEIM 5, 0 22 71/5 28 18. G

Software: Programme in Turbo-Pascal: (CP/M) Ahnenliste, Aktienverwaltung, Lernprogramme; Lottozahlenprogramme mit 1600 gespeicherten Zahlenreihen von 1955 bis aktuell. 8"-Format. B. Schöps, Stauffenbergstr. 9, 6940 Weinheim.

SAM-68K Suche Erfahrungs- und Programmtausch. Wer hat CAD und WordStar? Suche auch Hardware. P. Decker, Pf. 60 51, 2300 Kiel.

Firmenverzeichnis zum Anzeigenteil

acs, Detmold 89
adcomp, Datensysteme, München 81
ASC, Aachen 125
Atari, Raunheim 11

basys, Puchheim 131
BNT Computer, Stuttgart 122
Bohne & Speidel, Pforzheim 128
Brother, Bad Vilbel 12, 13
BS-Elektronik, Frankfurt 112

ccp datentechnik, Hamburg 125
CCP-Software, Marburg 126
CID, Hamburg 126
city elektronik, Dortmund 95
C + M Meyer, Viersen 136
ComFood, Münster 69
Computer Büromaschinen Service, Steinfurt 127
Computercentrum Hahn, Hemer .. 132
Conitec, Darmstadt 95
CSF, Bielefeld 110
CTK, Ransbach 38
CTK Computer, Bergisch-Gladbach 134

Data Becker, Düsseldorf 15
Decker, Nürnberg 75
Dederichs, Hattingen 135
Dobbertin, Brühl 121
DOBERT, Hamburg 75
Dorsch, Markt Igensdorf 95
DV Büromaschinenhandel, Schmoll, Lüchow 123

EBG, Schanuel, Darmstadt 129
ebh computer, Lüdinghausen 97
ECD, Zolling 21
Egermeier, Nürnberg 133
ELSA, Aachen 129
ES, Ismaning 75

Focus, Hannover 24
FRANK & BRITTING, Forst 125
Franzis-Verlag, München 74

GAS, Köln 43
Graphtec, Herrsching 20
GWK, Herzogenrath 22

Habersaat, Türkenfeld 18
Habersetzer, Polling 121
Hansa, Wilhelmshaven 129
HANTAREX, Altenkirchen 128
Heimsoeth, München 17
Hetron, München 122
Hobby-tronic, Dortmund 119
Hösch, Düsseldorf 135
HSP, Münster 131
HW Elektronik, Hamburg 106

Industry Computer, Bochum 133
INTUS, CH-Astano 135
isert, Eiterfeld 23
IST, Friedberg 116

Jeschke, Kelkheim 131, 134
Juki, Hamburg 140

Kayser, Braunschweig 129
Kirchner, Duisburg 130
Klicman, Köln 134
Köller, Schieder-Schwalenberg ... 132
KWEM, Göttingen 97

LECH-TECHNICS, Kerpen-Türnich 110

MAIL-SHOP, Hamburg 133
MARFLOW computing, Hannover 41
Mathes, Laer 25
MAYON, Germering 133
McGraw-Hill, Hamburg 125
Melchers, Bremen 19
Meyer, Hannover 129
Micromint, Erkrath 16
Microprocessing, München 63
Milde, München 75
Miro, Braunschweig 95
Mohwinkel, Leverkusen 131

NUCLEAR INTERFACE, Münster . 121

pc-plus, München 65
P.C. Quelle, Düsseldorf 124
Pfotenhauer, Achern 14, 26
Phytec, Mainz 125
Piper & Partner, München 121
PLANTRON, Bad Homburg 2
Proteus, Karlsruhe 136
PS-Computervertrieb, Düsseldorf 127
Pyramid, Freiburg 136

RAFI, Ravensburg 25
Rail, Babenhausen 135
Ranfft, Haar 125
RATEV, Ratingen 112, 130
REED, Herne 129
resco, Augsburg 43
Röckrath, Aachen 131
Roos, Kleve 59
Rose, Gladbeck 116
Rost, Springe 127

Segor electronics, Berlin 133
SIG Computer, Braunfels-Altenkirchen 136
Software Systems, Hannover 124
SONIC, Saarbrücken 129
Sprengel, Hannover 127
Springmann, Hannover 131
SYNELEC, München 9
Sypplie, Rastatt 127
SYSTEC, Münster 127
Szeredy Computing, Rosenheim . 65

Schmidtke, Aachen 131
Schwarz & Müller, Stephanskirchen 125

Taxan, ... 19
Tesco, Wiesentheid 123
THOMATRONIC, Rosenheim 133
top kit, Geretsried 27
Triebner, Nürnberg 33

ueding, Menden 133

Völkner, Braunschweig 121

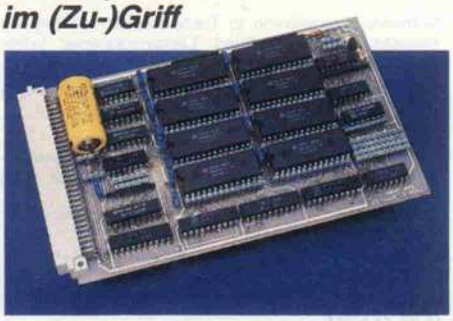
Weber, Würzburg 65
Wenzel, Niedernberg 134

Zacher, Irrel 43

In der nächsten

unter anderem

256 KByte dauerhaft im (Zu-)Griff



So bequem das Arbeiten mit einer RAM-Floppy auch ist, als lästiger Aufwand verbleibt doch, daß man sie erst initialisieren und von der 'lahmen' Floppy laden muß. Demgegenüber ist eine EPROM-Floppy in 'Null Komma nichts' einsatzfähig und hält wichtige Programme crash-sicher dauerhaft parat. Nur der Aufwand für die Programmierung der EPROMs ist normalerweise recht groß. Bei der c't-EPROM-Floppy hingegen lassen sich die EPROMs einfach auf der Karte mit einem speziellen PIP-Programm programmieren. Man kann die Karte aber auch mit CMOS-RAMs bestücken und erhält so eine gepufferte CMOS-RAM-Floppy, die alle Vorteile von EPROM, RAM und Floppy vereinigt.

IFC-Karte am Apple II

Leider unterstützt der Apple II 'von Haus aus' nur zwei 35-Track-Laufwerke, die sich weder durch große Speicherkapazität noch durch umwerfende Schnelligkeit auszeichnen. Was liegt also näher, als den Intelligenten Floppy-Controller (IFC) aus c't 8/85 auch am Apple zu betreiben. Der Z80-Prozessor, die DMA-Einheit und das intelligente Track-Buffering der IFC-Karte beschleunigen wirksam den Zugriff auf den Massenspeicher und entlasten den Hauptrechner.

Das Betriebssystem des Atari ST

. . . heißt GEM. Oder TOS? Oder CP/M-68K? Oder GEMDOS? Oder gar MS-DOS? Oder wieoderwas? Seit den ersten Ankündigungen von Atari machen vielerlei Gerüchte über dessen Betriebssystem die Runde. Daß da GEM war, wußte jeder; aber was liegt drunter und wie ist es aufgebaut? Ist man auf ewig zur Arbeit mit Mäusen und Fenstern verdammt, oder kann man auch mit dem Atari kommandoorientiert (und somit, wie manche meinen, vernünftig) arbeiten? Und wenn schon GEM: wie, bitte, macht man eigene GEM-Programme? Diese Fragen wollen wir in einer Artikelserie aufgreifen, deren erster Teil Sie in der nächsten c't erwartet.

Im Test: HRG-Karte für PCs

Daß man Farbgrafik nicht nur für Spiele, sondern auch für ernsthafte PC-Anwendungen benötigt, hat sich inzwischen weitgehend herumgesprochen. Die Vielzahl brauchbarer Software-Produkte, etwa auf den Gebieten des 'Computer Aided Design (CAD)' und der Präsentations-Grafik, leidet freilich unter den Schwächen, die der (IBM-)Industriestandard hardwareseitig aufweist. Kein Wunder, daß es inzwischen auch Zusatzkarten für eine wirklich hochauflösende Farbgrafik gibt. Wir haben uns eine 'echte' HRG-Karte für den IBM PC auf dem Prüfstand vorgenommen, die nicht nur 1 Million Bildpunkte mit 16 Farben pro Punkt, sondern auch eine Menge eigene 'Intelligenz' aufzubieten hat.

Heft 5/86 erscheint am 17. April 1986
Änderungen vorbehalten

Das bringen

INPUT 64 — DAS ELEKTRONISCHE MAGAZIN — Infos · News · Programme · Unterhaltung · Tips

INPUT 4/86 – ab 14. 4. 1986 am Kiosk

LISP64: Implementation der Ursprache der Künstlichen Intelligenz auf dem C64 * Der große Preis: das beliebte Fernsehquiz mit 150 Fragen für vier Kandidaten * Rechen-Tool: Hilfsprogramm zur Einbindung verschiedener Formeln in BASIC-Programme * LIFE: Simulations-Spiel * Mathe mit Nico: Zinsrechnung * u.v.a.m.

elrad — magazin für elektronik

elrad 4/86 – ab 24. 3. 1986 am Kiosk

Bauanleitung: Modularer Röhrensound – Bühnenverstärker EXPERIENCE * Grundlagen: FET-Schaltungspraxis * Bauelemente-Report: SMDs – Auf die Platte, fertig, los! * Grundlagen: Linkwitz-Filter * Video: Spectrum als Farbbalkengenerator * Bericht: Musikmesse Frankfurt * u.v.a.m.

Impressum:

c't Magazin für Computertechnik
Verlag Heinz Heise GmbH
Bissendorfer Straße 8, 3000 Hannover 61
Postanschrift: Postfach 61 04 07
3000 Hannover 61
Ruf (05 11) 5 35 20

technische Anfragen nur freitags 9.00—15.00 Uhr

Postscheckamt Hannover, Konto-Nr. 93 05-308
Kreissparkasse Hannover, Konto-Nr. 000-019968
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Andreas Burgwitz (stellvertretender Chefredakteur)
Dipl.-Ing. Detlef Grell
Johannes Assenbaum
Andreas Stiller
Manfred Bertuch
Ines Wurm

Ständige Mitarbeiter:
Dipl.-Ing. Rolf Keller
Dipl.-Ing. Eberhard Meyer
Dipl.-Ing. Eckart Steffens
Dipl.-Ing. Kurt Werner
Peter Rosenbeck, MA
Peter Glasmacher

Redaktionsassistenz: Thomas Nießen, Martina Klie

Technische Assistenz: Hans-Jürgen Berndt

Technische Zeichnungen: Marga Kellner

Bestellwesen: Christine Koop

Vertrieb: Anita Kreutzer-Tjaden

Anzeigen:
Wolfgang Penseler (Anzeigenleiter)
Gerlinde Donner (Disposition)
Es gilt die Anzeigenpreisliste 2 vom 1. 9. 1984

Redaktion, Anzeigenverwaltung:
Verlag Heinz Heise GmbH
Postfach 61 04 07
3000 Hannover 61
Ruf (05 11) 5 35 20

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Satz:
CW Niemeyer GmbH & Co KG Hameln
Ruf (0 51 51) 20 00

Druck:
Druckhaus Dierichs Kassel
Frankfurter Straße 168, 3500 Kassel

c't erscheint monatlich.
Einzelpreis DM 6,50, öS 57,–, sfr 6,50, hfl 7,40

Das Jahresabonnement kostet DM 71,– inkl. Versandkosten + MwSt., DM 83,– inkl. Versand (Ausland, Normalpost), DM 104,– inkl. Versand (Ausland, Luftpost).

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz) und Abonnementverwaltung:
Verlagsunion Zeitschriften-Vertrieb
Postfach 57 07
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Textteil: Christian Persson
Anzeigenteil: Wolfgang Penseler
beide Hannover, Bissendorfer Straße 8,
3000 Hannover 61

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Die geltenden gesetzlichen und postalischen Bestimmungen bei Erwerb, Errichtung und Inbetriebnahme von Sende- und Empfangseinrichtungen sind zu beachten.

Die gewerbliche Nutzung, insbesondere der Schaltpläne und gedruckten Schaltungen, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein.

Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Nachdruck nur mit Genehmigung des Verlages. Mit Übergabe der Manuskripte und Bilder an die Redaktion erteilt der Verfasser dem Verlag das Exclusivrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte kann keine Haftung übernommen werden.

Sämtliche Veröffentlichungen in ***c't*** erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany
© Copyright 1986 by Verlag Heinz Heise GmbH

ISSN 0724-8679

Titelidee: ***c't***
Titelfoto:
Zimmermann, Hannover

Software

c't-Programme

Dieses Angebot bezieht sich auf c't-Veröffentlichungen. Eine zusätzliche Dokumentation oder Bedienungsanleitung ist, soweit nicht anders angegeben, im Lieferumfang nicht enthalten. Eine Fotokopie der zugrundeliegenden Veröffentlichung können Sie unter Angabe der Programmnummer bestellen. Jede Kopie eines Beitrags kostet 5 DM, unabhängig vom Umfang. Eine Gewähr für das fehlerfreie Funktionieren der Programme kann nicht übernommen werden. Änderungen, insbesondere Verbesserungen, behalten wir uns vor.

Nr.	Programm	Datenträger	Preis
S831241	Terminal-Betriebsprogramm	EPROM (2732)	25 DM
S831244	Terminal-Betriebsprogramm V2.0 (für Terminal-Versionen A und B) inkl. Assembler-Listing	EPROM (2732)	35 DM
	Update (bei Rücksendung des vom Heise-Software-Service bezogenen EPROMs von V1.x), inkl. Assembler-Listing		20 DM
S831245	Assembler-Listing getrennt	Listing	12 DM
S831242	Terminal-Zeichensatz ZS0	EPROM (2732)	25 DM
S831243	Zeichensatz ZS1 (deutsch)	EPROM (2732)	25 DM
S8507116	c't 86 Monitor V2.0 inkl. Assembler-Source-Listing (für Farbgrafikkarte, IFC und Unicard)	2 EPROMs (2764)	75 DM
S840172	65C02-Assembler in FORTH	5¼-Zoll-Floppy (Apple)	15 DM
S840232	Energiekostenberechnung für Strom und Gas mit ZX81	Kassette	5 DM
S840324	Kfz-Kostenanalyse mit Video Genie	Kassette	5 DM
S840349	SPRITE-Editor für C64 (3 Versionen: Floppy-, Kassettenbetrieb und schnelles Maschinensprache-Programm)	Kassette (C64)	5 DM
S840580	Polygonberechnung	Kassette (ZX81)	5 DM
S840728	SET-65-Betriebsprogramm	EPROM (2764)	45 DM
S840729	SET-65-Dokumentation Ergänzung zum Handbuch '6502/65C02-Maschinensprache'	Listing	6 DM
S840792	GRIP-1-Betriebsprogramm mit Programmbeschreibung (siehe auch Platinen-Service)	EPROM (27128)	149 DM
S840826	PROF-80-Monitorprogramm V1.6 mit Source Listing V1.3 (siehe auch Platinen-Service)	EPROM (2764)	79 DM
S840827	PROF-80-Monitorprogramm im 200-ns-EPROM (6 MHz)	EPROM (2764)	89 DM
S840828	PROF-80-Monitorprogramm Source Code V1.3	Listing	15 DM
S840829	PROF-80-BIOS für CP/M 2.2	Listing	8 DM
S840636	Grafik-Tuning (Grafik-Programme für Apple II)	5¼-Zoll-Floppy	15 DM
S840881	CEPAC-65 als bidirektionales Interface für Brother CE 50/60	EPROM (2732)	25 DM
S841085	Cross Reference	5¼-Zoll-Floppy (Apple)	15 DM
S850154	Disassembler für ZX81, Disassembler für ZX Spectrum	Kassette (ZX81 und Spectrum)	5 DM
S8502103	CP/M 3.0 BIOS-Source-Listing für PROF-80	Listing	13 DM
S850332	Typenrad-Terminal mit Komfort (Neues Betriebsprogramm für TA SE 1005)	3 EPROMs (2716)	59 DM
S850333	Assembler-Listing dazu	Listing	10 DM
S8503104	SETFORTH — EPROM-Version SET-Betriebsprogramm inkl. Disassembler, FORTH-Compiler mit 65C02-Assembler, inkl. Glossary	EPROM (27128)	98 DM
S8503105	FORTH-Compiler mit 65C02-Assembler Kassettenversion, inkl. Glossary	Kassette (SuperTape)	59 DM
S8503106	SETFORTH, Kommentiertes Listing	Listing	29 DM
S8504110	MONALISA (EPAC-95-Monitor) inkl. Listing der Einsprungadressen	EPROM (2764)	59 DM
S850566	Klang-Computer-Betriebssoftware (alle Programme aus c't 12/84 bis 6/85)	Diskette (Apple)	25 DM
S850543	RAM-Disk-Treiber für Apple II	Diskette (Apple)	15 DM
S850683	Disk-Doktor für Apple II	Diskette	15 DM
S850668	C64-Treiberprogramm für CE 50/60	Diskette	15 DM
S8506112	Monitor für ZX81	Kassette	7 DM
S850676	Spooler-Programm	EPROM (2716)	25 DM
S850779	Verbessertes C64-Betriebsprogramm mit deutscher Tastenbelegung, umschaltbar auf Original	EPROM (27128)	59 DM
S850774	Wie S850779, zusätzlich Treibersoftware für IEEE-488-Schnittstelle	EPROM (27128)	69 DM
S850780	C64-Zeichensätze (original/deutsch)	EPROM (2764)	45 DM
S850766	G-Text für Apple II	Diskette	15 DM
S851042	Emulator 8080 für V-Chips	Diskette (IBM-CP/M86)	15 DM
S8511110	Fridolin-Spiel für ZX Spectrum	Kassette	7 DM
S851188	Funktionsplotter für Apple II	Diskette	35 DM
S860257	Apple-Zeichensätze (dt./ASCII, mager/Fett) für 80-Zeichenkarten	EPROM (2764)	25 DM
S860356	Sprachaufzeichnung u. -wiedergabe	Kassette (CPC)	7 DM
S860464	Lohn- und Einkommenssteuer '85	Kassette (CPC)	7 DM
SuperTape			
S840423	SuperTape für ZX81 (Basisroutinen, Betriebsprogramm und Kaltstart-Lader im ZX81-Format)	Kassette	5 DM
S840587	SuperTape für VC-20 und C64	Kassette	5 DM
S840733	SuperTape für Apple (incl. Source)	Diskette	15 DM
S840734	SuperTape für Apple	Kassette	5 DM
S850245	SuperTape für cbm 3000/4000/8000 (inkl. Source)	Diskette (4040)	15 DM
S850246	SuperTape für cbm 3000/4000/8000 (inkl. Source)	Diskette (8050)	15 DM
S8411112	SuperTape für CP/M-Rechner (Z80), Assembler-Listing	Listing	6 DM
S8505100	SuperTape für TRS80	Kassette	7 DM
S850890	SuperTape für Sharp MZ700	Kassette	7 DM
S850978	SuperTape für CPC 464	Kassette	7 DM
S851176	SuperTape für CPC 464/664 (CP/M)	Kassette	7 DM
S860282	SuperTape für C16/116, plus/4	Kassette	7 DM

S840001 Spectrum-Sammelkassette 1 **12,80 DM**

Die vier beliebtesten Spectrum-Programme aus c't 1984:

- SuperTape (mit Kaltstart-Lader und Betriebsprogramm)
- 'Platinen-Layout' (Experimentierprogramm zur Leiterplatinen-Entflechtung)
- Lohnsteuerberechnung
- Farmer (Gartenplanung am Bildschirm)

Programmbibliothek

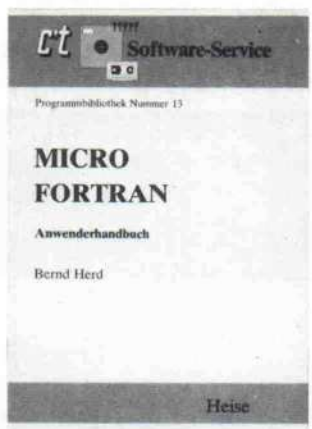
OTHELLO
(für Apple mit Pascal)
Das Strategiespiel Othello (Reversi) in einer schnellen Pascal-Version. Drei Spielstärken sind einstellbar. Das Handbuch enthält das Listing mit sehr ausführlicher Beschreibung und ist deshalb besonders interessant für Pascal-Anfänger.
Diskette (5¼-Zoll) mit Handbuch 30,— DM

Für Schneider CPC, Spectrum (48 K), TRS-80:
SUPERMON
Komfortabler Monitor/Debugger mit allen professionellen Features: Listen, Modifizieren, Testen von Maschinenprogrammen mit Breakpoints und Single Step. Z80-Disassembler, Fill-, Such-Funktionen. Eingabe wahlweise dezimal oder hexadezimal.
Bei Bestellung unbedingt Rechnertyp angeben!
Kassette mit Handbuch 39,— DM

Unser Bestseller:

AFORTH II
FORTH mit 65C02-Assembler
(für Apple und Apple-kompatible Computer mit Diskettenlaufwerk)
Das Programm enthält neben einem FORTH-Compiler nach dem FORTH-79-Standard einen zeilenorientierten Editor und einen Assembler für den erweiterten Befehlssatz der CMOS-CPU R65C02. Wenn das System mit einer 80-Zeichen-Karte ausgestattet ist, steht zusätzlich ein komfortabler Screen Editor zur Verfügung.
In 64-KByte-Systemen wird FORTH in die Language-Karte geladen und belegt den Adreßbereich (H) D000...F7FF. Die Transient Program Area (TPA) beginnt bei (H) 5000, so daß für High-Resolution-Anwendungen noch eine Seite frei bleibt. Bei anderen Systemen wird FORTH ab (H) 5000 geladen. Es steht dann mehr als 10 KByte Speicherraum für Anwenderprogramme zur Verfügung — wesentlich mehr als bei herkömmlichen FORTH-Systemen.
Der Compiler wird auf einer Diskette (Format: Apple Standard) geliefert, deren Rückseite das Source Listing des Assemblers und des Editors sowie nützliche Utilities wie einen FORTH-Decompiler und einen Textformatierer enthält. Es ist geplant, nach Festlegung des FORTH-83-Standards ein Anpassungsprogramm anzubieten.
Diskette mit Handbuch 98,— DM
Zwei Disketten (single sided) mit Handbuch 113,— DM

MICRO FORTRAN
(für TRS 80, Video Genie)
Micro Fortran ist ein Fortran-System für den TRS-80/Video Genie mit mindestens 16 K RAM und benötigt **keine** Diskettenstation. Da Fortran eine sehr umfangreiche Sprache ist und der Micro Fortran schon ab 16 K RAM arbeiten soll, enthält Micro Fortran nicht alle Möglichkeiten von Fortran IV. Trotzdem versteht das System die wichtigsten Fortran-Befehle, beherrscht Realzahlenverarbeitung und hat einen bequemen, bildschirmorientierten Editor. Im Vergleich zu BASIC ist Fortran wesentlich schneller, strukturierte Programmierung mit Unterprogrammen ist einfacher usw. Nachteilig ist allerdings, daß das compilierte Programm zwar sehr viel schneller ist als ein BASIC-Programmm, aber dafür auch wesentlich mehr Speicherplatz verbraucht. Außerdem muß für Fortran immer der Quelltext UND das Objektprogramm im Speicher stehen.
Das gesamte Fortran-System einschließlich Editor und Laufzeitsystem benötigt knapp unter 8 K Byte, es bleibt dem Benutzer also selbst bei nur 16 K noch genügend Platz, um einfache Programme zu schreiben.
Das Handbuch enthält eine Einführung in den Umgang mit FORTRAN und eine ausführliche Beschreibung aller unter MICRO FORTRAN verfügbaren Befehle.
Kassette und Handbuch 70,— DM
Diskettenversion 80,— DM



MICRO FORTRAN für ZX Spectrum (48 K)
Kassette und Handbuch 70,— DM

MICRO FORTRAN für Schneider CPC 464
Kassette und Handbuch 70,— DM

MICRO FORTH
(für TRS80, Model 1, und Video Genie)
MICRO FORTH ist ein ca. 8 KByte umfassender FORTH-Compiler für den Betrieb mit Kassettenrecorder. Auf der Kassette sind außerdem ein Editor und ein komfortabler Makro-Assembler (unter FORTH) enthalten. Das ausführliche Handbuch umfaßt neben der Beschreibung aller Befehle eine Anzahl von Programmbeispielen.
Kassette und Handbuch 70,— DM

c't-Klangcomputer
Sound Samples (Studioaufnahmen) für das DSM im EPROM

Diskette Vol. 1 (Apple II)	35 DM
Diskette Vol. 2 (Apple II)	35 DM
Kassette Vol. 1 + Vol. 2 (SuperTape)	69 DM
EPROM (Typen 2716...27128), je Instrument	25 DM

Eine Kurzbeschreibung der verschiedenen Klänge erhalten Sie gegen Zusendung eines rückadressierten Freiumschlages.

CP/M 86 für IBM PC
(mit englischer Dokumentation)
227,— DM

TurboGraf
Grafik-Paket für Apple II mit Turbo-PASCAL (läuft mit CP/M-Versionen ohne Bank Switching), inklusive Source. Neu: Jetzt auch für Turbo-Pascal 3.0
5¼-Zoll-Floppy (Apple) 69,— DM
Update für Besitzer der älteren Version, die nur mit Turbo-Pascal 2.x läuft, bei Einsendung der Originaldiskette 15,— DM

PROMMER80-Software
Betriebsprogramm zur menügesteuerten Programmierung aller gängigen EPROM-Typen (siehe c't 2/85)
8-Zoll-Floppy (IBM-Standardformat) 49,— DM

PROMMER80-Software
für Schneider CPC (siehe c't 2/86)
Kassette 39,— DM

PROMMER-86-Software
wie PROMMER 80, angepaßt an den c't 86 (siehe c't 12/85).
Version A 49 DM
(CP/M-86, 5,25-Zoll, IBM-PC-Lieferformat)
Version B 49 DM
(PC-DOS, 5,25-Zoll, IBM-PC-Lieferformat)

Netzwerkanalyse
(Beschreibung siehe c't 12/85)

für C64, CBM 3000/4000/8000	Kassette 25 DM
für C64, C16, Plus/4	Diskette 39 DM
für Apple II	Diskette 39 DM

NEU:
INPUT-64-BASIC-Erweiterung
in zwei 2764er-Eproms für die C64-EPROM-Bank. Über 40 neue Befehle und SuperTape Dll. 49,— DM
Bits & Bytes im Video-Chip
Der INPUT-64-Kurs über den Video-Chip im Commodore 64

Diskette	24,80 DM
Kassette	17,80 DM

Die Handbücher zu den Programmen, soweit in der Anzeige aufgeführt, sind zum Preis von je 5 DM (inklusive Porto) getrennt erhältlich. Bei einer Bestellung des Programms wird der Betrag angerechnet. (Bitte vermerken Sie auf Ihrer Bestellung 'Ohne Handbuch'.)

So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck über die Bestellsumme zuzüglich DM 3,— (für Porto und Verpackung) bei oder überweisen Sie den Betrag auf eines unserer Konten.

Schecks werden erst bei Lieferung eingelöst. Wir empfehlen deshalb diesen Zahlungsweg, da in Einzelfällen längere Lieferzeiten auftreten können.

Bankverbindungen:
Postscheckamt Hannover, Kt.-Nr. 93 05-308
Kreissparkasse Hannover, Kt.-Nr. 000-019968
(BLZ 250 502 99)

Ihre Bestellung richten Sie bitte an:

Verlag Heinz Heise GmbH
Postfach 61 04 07
3000 Hannover 61

JUKI. Für hellere Ideen...

FARBE

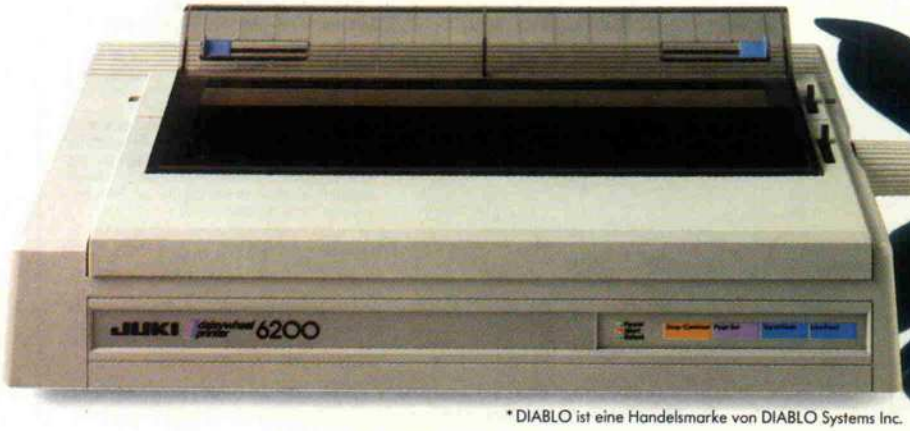
Der Farb-Matrix-Drucker JUKI 5520 liefert Ihnen sieben Farben (für Spreadsheet besonders ideal) zum Schwarz/Weiß-Preis! Und noch mehr: Durch umlegen des Dipschalters erreicht man Kompatibilität zum Epson* JX-80 und zum IBM*-Farbgrafikdrucker. Der JUKI 5520 druckt sowohl im Near Letter Quality – als auch im Grafikmodus und verfügt über eine eingebaute Papierzufuhr. Und Bidirektionaldruck schafft er mit einer Geschwindigkeit von 180 ZPS. Der JUKI 5520 – schon wieder eine glänzende Idee von uns.

* Epson ist eine Handelsmarke von Epson.
* IBM ist eine Handelsmarke der IBM Corporation.
Ausdruck erzeugt unter Benutzung von Colorshop DATA FANT.

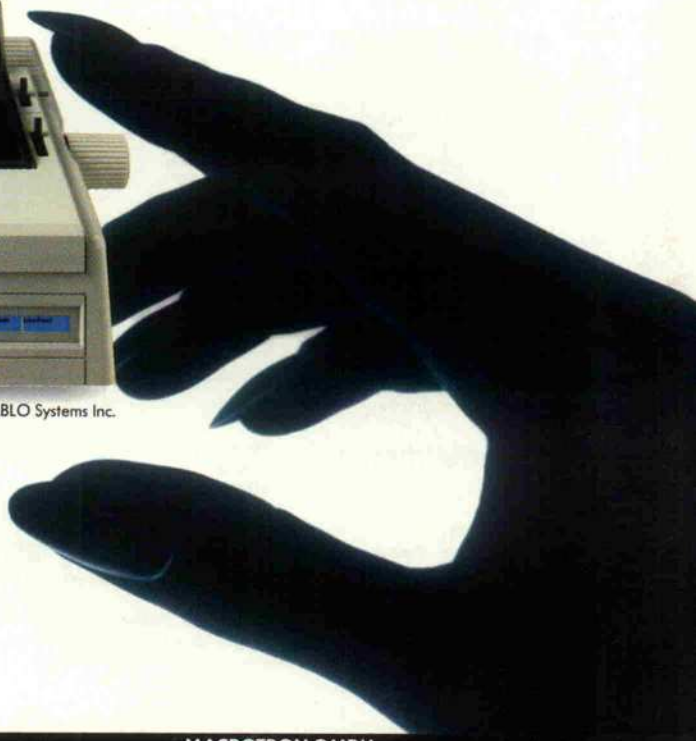
...und hellere Köpfchen

Der JUKI 6200 dagegen ist ein preisgünstiger und rapider Typenraddrucker, der sämtlichen Textverarbeitungsaufgaben voll gewachsen ist. Die Druckqualität ist bedeutend schärfer gestochen und gehobener als Full Letter Quality, und dank seiner extra-breiten 16-Zoll-Walze passen ihm auch großformatige Papierbogen. Das Beste an ihm jedoch ist, daß er mit seinem standardmäßigen, 96 Zeichen umfassenden DIABLO* Typenrad eine Höchstgeschwindigkeit von 32 ZPS hinlegt. Der JUKI 6200 – ein schneller Denker und flotter Drucker.

32 CPS



* DIABLO ist eine Handelsmarke von DIABLO Systems Inc.



JUKI

Technology true to type

JUKI (EUROPE) GMBH
Eiffestraße 74 · 2000 Hamburg 26
Tel.: (0 40) 2 51 20 71-73 · Telex: 2 163 061 (JKI D)
Fax: (0 40) 2 51 27 24

VERTEILER

MVB GMBH
Brüder-Grimm-Straße 5 · 6408 Ebersburg-Weyhers Kreis Fulda
Telefon 0 66 56/10 56, 10 57 · Abt. Service 10 58 · Telex 4 9 792

MIRWALD ELECTRONIC
Fasanenstraße 8b · 8025 Unterhaching bei München
Telefon 0 89/6 11 12 24 und 6 11 20 40, Telex 5 213 476

MACROTRON GMBH
Stahlgruberring 28 · 8000 München 82
Telefon 0 89/42 08-100 · Telex 5 29 448 mato